1. import 'package:flutter/material.dart';
2. import 'package:intl/intl.dart';
3. import 'package:shared\_preferences/shared\_preferences.dart';
4. import 'package:provider/provider.dart';
5. import 'package:google\_fonts/google\_fonts.dart';
6. import 'package:url\_launcher/url\_launcher.dart';
7. import 'package:fl\_chart/fl\_chart.dart';
8. import 'dart:convert';
9. import 'package:pdf/pdf.dart';
10. import 'package:pdf/widgets.dart' as pw;
11. import 'package:printing/printing.dart';
12. // Text size provider for managing text scaling
13. class TextSizeProvider extends ChangeNotifier {
14. double \_textScaleFactor = 1.0;
15. double get textScaleFactor => \_textScaleFactor;
16. void increaseTextSize() {
17. if (\_textScaleFactor < 1.3) {
18. \_textScaleFactor += 0.1;
19. notifyListeners();
20. }
21. }
22. void resetTextSize() {
23. \_textScaleFactor = 1.0;
24. notifyListeners();
25. }
26. void decreaseTextSize() {
27. if (\_textScaleFactor > 0.7) {
28. \_textScaleFactor -= 0.1;
29. notifyListeners();
30. }
31. }
32. }
33. void main() async {
34. WidgetsFlutterBinding.ensureInitialized();
35. final authService = AuthService();
36. await authService.loadAuthState();
37. runApp(
38. MultiProvider(
39. providers: [
40. ChangeNotifierProvider(create: (\_) => authService),
41. ChangeNotifierProvider(create: (\_) => TextSizeProvider()),
42. ChangeNotifierProvider(create: (\_) => AppointmentService()),
43. ChangeNotifierProvider(create: (\_) => FeedbackService()),
44. ],
45. child: GlobalCareApp(),
46. ),
47. );
48. }
49. class GlobalCareApp extends StatelessWidget {
50. @override
51. Widget build(BuildContext context) {
52. return Consumer<TextSizeProvider>(
53. builder: (context, textSizeProvider, child) {
54. return MaterialApp(
55. title: 'Global Care Medical Center',
56. theme: ThemeData(
57. primarySwatch: Colors.indigo,
58. fontFamily: GoogleFonts.poppins().fontFamily,
59. scaffoldBackgroundColor: Color(0xFFF8FAFC),
60. textTheme: TextTheme(
61. headlineMedium: TextStyle(
62. fontSize: 28,
63. fontWeight: FontWeight.w600,
64. color: Color(0xFF1F2937),
65. ),
66. titleLarge: TextStyle(
67. fontSize: 18,
68. fontWeight: FontWeight.w500,
69. color: Color(0xFF1F2937),
70. ),
71. bodyMedium: TextStyle(
72. fontSize: 16,
73. fontWeight: FontWeight.w400,
74. color: Color(0xFF6B7280),
75. ),
76. labelLarge: TextStyle(
77. fontSize: 14,
78. fontWeight: FontWeight.w500,
79. color: Colors.white,
80. ),
81. ),
82. appBarTheme: AppBarTheme(
83. backgroundColor: Colors.white,
84. elevation: 0,
85. shadowColor: Colors.black12,
86. iconTheme: IconThemeData(color: Colors.indigo[900]),
87. ),
88. elevatedButtonTheme: ElevatedButtonThemeData(
89. style: ElevatedButton.styleFrom(
90. backgroundColor: Colors.indigo,
91. foregroundColor: Colors.white,
92. elevation: 3,
93. shadowColor: Colors.indigo.withOpacity(0.3),
94. shape: RoundedRectangleBorder(
95. borderRadius: BorderRadius.circular(24),
96. ),
97. padding: EdgeInsets.symmetric(horizontal: 24, vertical: 12),
98. textStyle: TextStyle(
99. fontFamily: GoogleFonts.poppins().fontFamily,
100. fontSize: 16,
101. fontWeight: FontWeight.w500,
102. ),
103. ),
104. ),
105. textButtonTheme: TextButtonThemeData(
106. style: TextButton.styleFrom(
107. foregroundColor: Colors.indigo,
108. backgroundColor: Colors.transparent,
109. shape: RoundedRectangleBorder(
110. borderRadius: BorderRadius.circular(24),
111. ),
112. padding: EdgeInsets.symmetric(horizontal: 16, vertical: 8),
113. minimumSize: Size(100, 36),
114. textStyle: TextStyle(
115. fontFamily: GoogleFonts.poppins().fontFamily,
116. fontSize: 16,
117. fontWeight: FontWeight.w500,
118. ),
119. ),
120. ),
121. outlinedButtonTheme: OutlinedButtonThemeData(
122. style: OutlinedButton.styleFrom(
123. foregroundColor: Colors.indigo,
124. side: BorderSide(color: Colors.indigo, width: 1.5),
125. shape: RoundedRectangleBorder(
126. borderRadius: BorderRadius.circular(24),
127. ),
128. textStyle: TextStyle(
129. fontFamily: GoogleFonts.poppins().fontFamily,
130. fontSize: 16,
131. fontWeight: FontWeight.w500,
132. ),
133. ),
134. ),
135. cardTheme: CardThemeData(
136. elevation: 4,
137. shape: RoundedRectangleBorder(
138. borderRadius: BorderRadius.circular(16),
139. ),
140. shadowColor: Colors.black12,
141. ),
142. ),
143. initialRoute: '/',
144. routes: {
145. '/': (context) => MainPage(),
146. '/appointment-receipt': (context) => AppointmentReceiptPage(),
147. },
148. debugShowCheckedModeBanner: false,
149. builder: (context, child) {
150. return MediaQuery(
151. data: MediaQuery.of(context).copyWith(
152. textScaler: TextScaler.linear(textSizeProvider.textScaleFactor),
153. ),
154. child: child!,
155. );
156. },
157. );
158. },
159. );
160. }
161. }
162. class MainPage extends StatefulWidget {
163. @override
164. \_MainPageState createState() => \_MainPageState();
165. }
166. class \_MainPageState extends State<MainPage> with TickerProviderStateMixin {
167. Widget \_buildNavItem(
168. BuildContext context,
169. String title,
170. int index,
171. bool isActive,
172. bool isMobile,
173. ) {
174. return GestureDetector(
175. onTap: () {
176. DefaultTabController.of(context)?.animateTo(index);
177. (context as Element).markNeedsBuild();
178. },
179. child: Text(
180. title,
181. style: GoogleFonts.poppins(
182. fontSize: 16,
183. fontWeight: isActive ? FontWeight.w600 : FontWeight.w400,
184. color: isActive ? Colors.indigo[700] : Colors.black,
185. ),
186. ),
187. );
188. }
189. void \_showAuthDialog(BuildContext context) {
190. showDialog(context: context, builder: (context) => AuthDialog());
191. }
192. @override
193. Widget build(BuildContext context) {
194. final authService = Provider.of<AuthService>(context);
195. final textSizeProvider = Provider.of<TextSizeProvider>(context);
196. return DefaultTabController(
197. length: 4,
198. initialIndex: 0,
199. child: Scaffold(
200. appBar: AppBar(
201. automaticallyImplyLeading: false,
202. title: LayoutBuilder(
203. builder: (context, constraints) {
204. final isMobile = constraints.maxWidth < 600;
205. return Row(
206. mainAxisAlignment: MainAxisAlignment.spaceBetween,
207. crossAxisAlignment: CrossAxisAlignment.center,
208. children: [
209. Row(
210. children: [
211. Icon(
212. Icons.favorite,
213. color: Colors.indigo[700]!,
214. size: isMobile ? 28 : 32,
215. ),
216. SizedBox(width: 8),
217. Text(
218. 'Global Care',
219. style: GoogleFonts.poppins(
220. fontWeight: FontWeight.w700,
221. color: Colors.indigo[900]!,
222. fontSize: isMobile ? 20 : 24,
223. ),
224. ),
225. ],
226. ),
227. if (isMobile) ...[
228. Row(
229. children: [
230. IconButton(
231. icon: Icon(
232. Icons.account\_circle,
233. color: Colors.indigo[700]!,
234. size: isMobile ? 24 : 28,
235. ),
236. onPressed: () {
237. if (authService.isLoggedIn) {
238. final tabController = DefaultTabController.of(
239. context,
240. );
241. if (tabController != null) {
242. tabController.animateTo(3); // Dashboard tab
243. (context as Element).markNeedsBuild();
244. }
245. } else {
246. \_showAuthDialog(context);
247. }
248. },
249. tooltip: authService.isLoggedIn
250. ? 'Profile'
251. : 'Login/Sign Up',
252. ),
253. IconButton(
254. icon: Icon(Icons.menu, color: Colors.indigo[700]!),
255. onPressed: () {
256. Scaffold.of(context).openDrawer();
257. },
258. ),
259. ],
260. ),
261. ] else ...[
262. Expanded(
263. child: Center(
264. child: Row(
265. mainAxisSize: MainAxisSize.min,
266. children: [
267. \_buildNavItem(
268. context,
269. 'Home',
270. 0,
271. DefaultTabController.of(context)?.index == 0,
272. isMobile,
273. ),
274. SizedBox(width: 24),
275. \_buildNavItem(
276. context,
277. 'Appointments',
278. 1,
279. DefaultTabController.of(context)?.index == 1,
280. isMobile,
281. ),
282. SizedBox(width: 24),
283. \_buildNavItem(
284. context,
285. 'Feedback',
286. 2,
287. DefaultTabController.of(context)?.index == 2,
288. isMobile,
289. ),
290. SizedBox(width: 24),
291. \_buildNavItem(
292. context,
293. 'Dashboard',
294. 3,
295. DefaultTabController.of(context)?.index == 3,
296. isMobile,
297. ),
298. ],
299. ),
300. ),
301. ),
302. Row(
303. children: [
304. if (!authService.isLoggedIn) ...[
305. Container(
306. decoration: BoxDecoration(
307. border: Border.all(
308. color: Colors.indigo[700]!,
309. width: 1,
310. ),
311. borderRadius: BorderRadius.circular(24),
312. ),
313. child: ElevatedButton(
314. onPressed: () => \_showAuthDialog(context),
315. child: Text(
316. 'Login',
317. style: TextStyle(
318. fontFamily: GoogleFonts.poppins().fontFamily,
319. fontSize: 14, // Reduced font size
320. fontWeight: FontWeight.w500,
321. color: Colors.indigo[700],
322. ),
323. ),
324. style: ElevatedButton.styleFrom(
325. backgroundColor: Colors.white,
326. shadowColor: Colors.transparent,
327. padding: EdgeInsets.symmetric(
328. horizontal: isMobile
329. ? 12
330. : 16, // Reduced padding
331. vertical: isMobile ? 6 : 8, // Reduced padding
332. ),
333. shape: RoundedRectangleBorder(
334. borderRadius: BorderRadius.circular(24),
335. ),
336. ),
337. ),
338. ),
339. SizedBox(width: isMobile ? 4 : 8),
340. Container(
341. decoration: BoxDecoration(
342. border: Border.all(
343. color: Colors.indigo[700]!,
344. width: 1,
345. ),
346. borderRadius: BorderRadius.circular(24),
347. ),
348. child: ElevatedButton(
349. onPressed: () => \_showAuthDialog(context),
350. child: Text(
351. 'Sign Up',
352. style: TextStyle(
353. fontFamily: GoogleFonts.poppins().fontFamily,
354. fontSize: 14, // Reduced font size
355. fontWeight: FontWeight.w500,
356. color: Colors.white,
357. ),
358. ),
359. style: ElevatedButton.styleFrom(
360. backgroundColor: Colors.indigo[700],
361. shadowColor: Colors.transparent,
362. padding: EdgeInsets.symmetric(
363. horizontal: isMobile
364. ? 12
365. : 16, // Reduced padding
366. vertical: isMobile ? 6 : 8, // Reduced padding
367. ),
368. shape: RoundedRectangleBorder(
369. borderRadius: BorderRadius.circular(24),
370. ),
371. ),
372. ),
373. ),
374. ] else ...[
375. Row(
376. children: [
377. CircleAvatar(
378. backgroundColor: Colors.indigo.withOpacity(0.1),
379. child: Icon(
380. Icons.person,
381. color: Colors.indigo[700]!,
382. size: isMobile ? 20 : 24,
383. ),
384. radius: isMobile ? 14 : 16,
385. ),
386. SizedBox(width: 8),
387. Text(
388. '${authService.username}',
389. style: GoogleFonts.poppins(
390. color: Colors.indigo[700]!,
391. fontSize: isMobile ? 14 : 16,
392. fontWeight: FontWeight.w500,
393. ),
394. ),
395. ],
396. ),
397. SizedBox(width: 8),
398. IconButton(
399. icon: Icon(
400. Icons.logout,
401. color: Colors.indigo[700]!,
402. ),
403. onPressed: () => authService.logout(),
404. ),
405. ],
406. ],
407. ),
408. ],
409. ],
410. );
411. },
412. ),
413. ),
414. drawer: LayoutBuilder(
415. builder: (context, constraints) {
416. final isMobile = constraints.maxWidth < 600;
417. if (!isMobile) return SizedBox.shrink();
418. return Drawer(
419. child: ListView(
420. padding: EdgeInsets.zero,
421. children: [
422. DrawerHeader(
423. decoration: BoxDecoration(color: Colors.indigo[700]!),
424. child: Row(
425. children: [
426. Icon(Icons.favorite, color: Colors.white, size: 32),
427. SizedBox(width: 8),
428. Text(
429. 'Global Care',
430. style: GoogleFonts.poppins(
431. color: Colors.white,
432. fontSize: 24,
433. fontWeight: FontWeight.w700,
434. ),
435. ),
436. ],
437. ),
438. ),
439. ListTile(
440. leading: Icon(Icons.home, color: Colors.indigo[700]!),
441. title: Text(
442. 'Home',
443. style: GoogleFonts.poppins(
444. fontSize: 16,
445. color: DefaultTabController.of(context)?.index == 0
446. ? Colors.indigo[700]
447. : Colors.black,
448. fontWeight: DefaultTabController.of(context)?.index == 0
449. ? FontWeight.w600
450. : FontWeight.w400,
451. ),
452. ),
453. onTap: () {
454. DefaultTabController.of(context)?.animateTo(0);
455. Navigator.pop(context);
456. (context as Element).markNeedsBuild();
457. },
458. ),
459. ListTile(
460. leading: Icon(Icons.event, color: Colors.indigo[700]!),
461. title: Text(
462. 'Appointments',
463. style: GoogleFonts.poppins(
464. fontSize: 16,
465. color: DefaultTabController.of(context)?.index == 1
466. ? Colors.indigo[700]
467. : Colors.black,
468. fontWeight: DefaultTabController.of(context)?.index == 1
469. ? FontWeight.w600
470. : FontWeight.w400,
471. ),
472. ),
473. onTap: () {
474. DefaultTabController.of(context)?.animateTo(1);
475. Navigator.pop(context);
476. (context as Element).markNeedsBuild();
477. },
478. ),
479. ListTile(
480. leading: Icon(Icons.feedback, color: Colors.indigo[700]!),
481. title: Text(
482. 'Feedback',
483. style: GoogleFonts.poppins(
484. fontSize: 16,
485. color: DefaultTabController.of(context)?.index == 2
486. ? Colors.indigo[700]
487. : Colors.black,
488. fontWeight: DefaultTabController.of(context)?.index == 2
489. ? FontWeight.w600
490. : FontWeight.w400,
491. ),
492. ),
493. onTap: () {
494. DefaultTabController.of(context)?.animateTo(2);
495. Navigator.pop(context);
496. (context as Element).markNeedsBuild();
497. },
498. ),
499. ListTile(
500. leading: Icon(Icons.dashboard, color: Colors.indigo[700]!),
501. title: Text(
502. 'Dashboard',
503. style: GoogleFonts.poppins(
504. fontSize: 16,
505. color: DefaultTabController.of(context)?.index == 3
506. ? Colors.indigo[700]
507. : Colors.black,
508. fontWeight: DefaultTabController.of(context)?.index == 3
509. ? FontWeight.w600
510. : FontWeight.w400,
511. ),
512. ),
513. onTap: () {
514. DefaultTabController.of(context)?.animateTo(3);
515. Navigator.pop(context);
516. (context as Element).markNeedsBuild();
517. },
518. ),
519. ],
520. ),
521. );
522. },
523. ),
524. body: TabBarView(
525. children: [
526. HomePage(),
527. AppointmentPage(),
528. FeedbackPage(),
529. DashboardPage(),
530. ],
531. ),
532. floatingActionButton: LayoutBuilder(
533. builder: (context, constraints) {
534. final isMobile = constraints.maxWidth < 600;
535. return Container(
536. padding: EdgeInsets.all(isMobile ? 6 : 8),
537. decoration: BoxDecoration(
538. color: Colors.white,
539. borderRadius: BorderRadius.circular(24),
540. boxShadow: [
541. BoxShadow(
542. color: Colors.black.withOpacity(0.1),
543. blurRadius: 8,
544. offset: Offset(0, 2),
545. ),
546. ],
547. ),
548. child: Row(
549. mainAxisSize: MainAxisSize.min,
550. children: [
551. \_buildFAB(
552. heroTag: 'decreaseText',
553. onPressed: textSizeProvider.decreaseTextSize,
554. tooltip: 'Decrease text size',
555. icon: Icons.text\_decrease,
556. isMobile: isMobile,
557. ),
558. \_buildFAB(
559. heroTag: 'resetText',
560. onPressed: textSizeProvider.resetTextSize,
561. tooltip: 'Reset text size',
562. icon: Icons.text\_fields,
563. isMobile: isMobile,
564. ),
565. \_buildFAB(
566. heroTag: 'increaseText',
567. onPressed: textSizeProvider.increaseTextSize,
568. tooltip: 'Increase text size',
569. icon: Icons.text\_increase,
570. isMobile: isMobile,
571. ),
572. ],
573. ),
574. );
575. },
576. ),
577. floatingActionButtonLocation: FloatingActionButtonLocation.endFloat,
578. ),
579. );
580. }
581. Widget \_buildFAB({
582. required String heroTag,
583. required VoidCallback onPressed,
584. required String tooltip,
585. required IconData icon,
586. required bool isMobile,
587. }) {
588. return IconButton(
589. icon: Icon(icon, color: Colors.indigo[700]!, size: isMobile ? 18 : 20),
590. onPressed: onPressed,
591. tooltip: tooltip,
592. padding: EdgeInsets.all(isMobile ? 6 : 8),
593. constraints: BoxConstraints(),
594. splashColor: Colors.blue[600]!,
595. highlightColor: Colors.blue[400]!,
596. );
597. }
598. }
599. class FeedbackPage extends StatefulWidget {
600. @override
601. State<FeedbackPage> createState() => \_FeedbackPageState();
602. }
603. class \_FeedbackPageState extends State<FeedbackPage> {
604. final \_formKey = GlobalKey<FormState>();
605. final feedbackController = TextEditingController();
606. double satisfactionRating = 3.0;
607. bool \_isSubmitting = false;
608. void \_submitFeedback() async {
609. if (!\_formKey.currentState!.validate()) {
610. return;
611. }
612. setState(() => \_isSubmitting = true);
613. final feedback = {
614. 'rating': satisfactionRating,
615. 'comment': feedbackController.text,
616. 'date': DateFormat('MMM d, yyyy').format(DateTime.now()),
617. 'username':
618. Provider.of<AuthService>(context, listen: false).username ?? 'Guest',
619. };
620. try {
621. await Provider.of<FeedbackService>(
622. context,
623. listen: false,
624. ).saveFeedback(feedback);
625. ScaffoldMessenger.of(context).showSnackBar(
626. SnackBar(
627. content: Text('Feedback submitted successfully!'),
628. backgroundColor: Colors.green,
629. behavior: SnackBarBehavior.floating,
630. ),
631. );
632. setState(() {
633. feedbackController.clear();
634. satisfactionRating = 3.0;
635. });
636. } catch (e) {
637. ScaffoldMessenger.of(context).showSnackBar(
638. SnackBar(
639. content: Text('Failed to save feedback: $e'),
640. backgroundColor: Colors.red,
641. behavior: SnackBarBehavior.floating,
642. ),
643. );
644. } finally {
645. setState(() => \_isSubmitting = false);
646. }
647. }
648. @override
649. Widget build(BuildContext context) {
650. return Consumer<AuthService>(
651. builder: (context, authService, child) {
652. return SingleChildScrollView(
653. child: Column(
654. children: [
655. LayoutBuilder(
656. builder: (context, constraints) {
657. final isMobile = constraints.maxWidth < 600;
658. return Container(
659. color: Color(0xFFF8FAFC),
660. padding: EdgeInsets.symmetric(
661. vertical: isMobile ? 32 : 48,
662. horizontal: constraints.maxWidth \* 0.05,
663. ),
664. child: Column(
665. children: [
666. // Header
667. Column(
668. children: [
669. Text(
670. 'Your Feedback Matters',
671. style: GoogleFonts.poppins(
672. fontSize: isMobile ? 26 : 30,
673. fontWeight: FontWeight.w700,
674. color: Color(0xFF1F2937),
675. ),
676. textAlign: TextAlign.center,
677. ),
678. SizedBox(height: 12),
679. Text(
680. 'Help us improve by sharing your experience.',
681. style: GoogleFonts.poppins(
682. fontSize: isMobile ? 15 : 16,
683. fontWeight: FontWeight.w400,
684. color: Color(0xFF6B7280),
685. ),
686. textAlign: TextAlign.center,
687. ),
688. ],
689. ),
690. SizedBox(height: isMobile ? 24 : 32),
691. // Feedback Form or Login Prompt
692. AnimatedOpacity(
693. opacity: 1.0,
694. duration: Duration(milliseconds: 500),
695. child: Card(
696. elevation: 6,
697. shape: RoundedRectangleBorder(
698. borderRadius: BorderRadius.circular(16),
699. ),
700. child: Container(
701. constraints: BoxConstraints(maxWidth: 600),
702. padding: EdgeInsets.all(isMobile ? 20 : 24),
703. decoration: BoxDecoration(
704. color: Colors.white,
705. borderRadius: BorderRadius.circular(16),
706. boxShadow: [
707. BoxShadow(
708. color: Colors.black.withOpacity(0.05),
709. blurRadius: 10,
710. offset: Offset(0, 4),
711. ),
712. ],
713. ),
714. child: !authService.isLoggedIn
715. ? \_buildLoginPrompt(isMobile)
716. : \_buildFeedbackForm(isMobile),
717. ),
718. ),
719. ),
720. // Added spacing before footer
721. SizedBox(height: isMobile ? 32 : 48),
722. ],
723. ),
724. );
725. },
726. ),
727. // Footer now included in the scrollable content
728. FooterSection(),
729. ],
730. ),
731. );
732. },
733. );
734. }
735. Widget \_buildLoginPrompt(bool isMobile) {
736. return Column(
737. mainAxisAlignment: MainAxisAlignment.center,
738. crossAxisAlignment: CrossAxisAlignment.center,
739. children: [
740. Icon(Icons.lock, size: isMobile ? 36 : 40, color: Colors.indigo[700]),
741. SizedBox(height: 16),
742. Text(
743. 'Sign in to share your feedback',
744. style: GoogleFonts.poppins(
745. fontSize: isMobile ? 18 : 20,
746. fontWeight: FontWeight.w600,
747. color: Color(0xFF1F2937),
748. ),
749. textAlign: TextAlign.center,
750. ),
751. SizedBox(height: 12),
752. Text(
753. 'Please login or create an account to provide your valuable feedback.',
754. style: GoogleFonts.poppins(
755. fontSize: isMobile ? 14 : 15,
756. color: Colors.grey[600],
757. ),
758. textAlign: TextAlign.center,
759. ),
760. SizedBox(height: 24),
761. ElevatedButton(
762. onPressed: () {
763. showDialog(context: context, builder: (context) => AuthDialog());
764. },
765. style: ElevatedButton.styleFrom(
766. backgroundColor: Colors.indigo[700],
767. foregroundColor: Colors.white,
768. padding: EdgeInsets.symmetric(
769. horizontal: isMobile ? 24 : 32,
770. vertical: isMobile ? 12 : 14,
771. ),
772. shape: RoundedRectangleBorder(
773. borderRadius: BorderRadius.circular(12),
774. ),
775. elevation: 2,
776. ),
777. child: Text(
778. 'Login / Sign Up',
779. style: GoogleFonts.poppins(
780. fontSize: isMobile ? 15 : 16,
781. fontWeight: FontWeight.w600,
782. ),
783. ),
784. ),
785. ],
786. );
787. }
788. Widget \_buildFeedbackForm(bool isMobile) {
789. return Form(
790. key: \_formKey,
791. child: Column(
792. crossAxisAlignment: CrossAxisAlignment.start,
793. children: [
794. Text(
795. 'Rate Your Experience',
796. style: GoogleFonts.poppins(
797. fontSize: isMobile ? 18 : 20,
798. fontWeight: FontWeight.w600,
799. color: Color(0xFF1F2937),
800. ),
801. ),
802. SizedBox(height: 12),
803. Row(
804. mainAxisAlignment: MainAxisAlignment.spaceBetween,
805. children: [
806. Text(
807. 'Satisfaction Level: ${satisfactionRating.toStringAsFixed(1)}',
808. style: GoogleFonts.poppins(
809. fontSize: isMobile ? 14 : 15,
810. color: Colors.grey[600],
811. ),
812. ),
813. Row(
814. children: List.generate(5, (index) {
815. return Icon(
816. index < satisfactionRating.floor()
817. ? Icons.star
818. : Icons.star\_border,
819. color: Colors.amber,
820. size: isMobile ? 24 : 28,
821. );
822. }),
823. ),
824. ],
825. ),
826. SizedBox(height: 8),
827. Slider(
828. value: satisfactionRating,
829. min: 1,
830. max: 5,
831. divisions: 4,
832. label: satisfactionRating.toStringAsFixed(1),
833. onChanged: (value) => setState(() => satisfactionRating = value),
834. activeColor: Colors.indigo[700],
835. inactiveColor: Colors.indigo[100],
836. ),
837. SizedBox(height: 24),
838. Text(
839. 'Your Feedback',
840. style: GoogleFonts.poppins(
841. fontSize: isMobile ? 18 : 20,
842. fontWeight: FontWeight.w600,
843. color: Color(0xFF1F2937),
844. ),
845. ),
846. SizedBox(height: 12),
847. AnimatedContainer(
848. duration: Duration(milliseconds: 300),
849. decoration: BoxDecoration(
850. borderRadius: BorderRadius.circular(12),
851. boxShadow: [
852. BoxShadow(
853. color: Colors.black.withOpacity(0.05),
854. blurRadius: 6,
855. offset: Offset(0, 2),
856. ),
857. ],
858. ),
859. child: TextFormField(
860. controller: feedbackController,
861. decoration: InputDecoration(
862. labelText: 'Share your thoughts',
863. hintText: 'Tell us about your experience or suggestions...',
864. border: OutlineInputBorder(
865. borderRadius: BorderRadius.circular(12),
866. borderSide: BorderSide.none,
867. ),
868. enabledBorder: OutlineInputBorder(
869. borderRadius: BorderRadius.circular(12),
870. borderSide: BorderSide.none,
871. ),
872. focusedBorder: OutlineInputBorder(
873. borderRadius: BorderRadius.circular(12),
874. borderSide: BorderSide(color: Colors.indigo[700]!, width: 2),
875. ),
876. errorBorder: OutlineInputBorder(
877. borderRadius: BorderRadius.circular(12),
878. borderSide: BorderSide(color: Colors.red[400]!, width: 1),
879. ),
880. focusedErrorBorder: OutlineInputBorder(
881. borderRadius: BorderRadius.circular(12),
882. borderSide: BorderSide(color: Colors.red[400]!, width: 2),
883. ),
884. filled: true,
885. fillColor: Colors.indigo.withOpacity(0.05),
886. labelStyle: GoogleFonts.poppins(
887. color: Colors.grey[600],
888. fontSize: isMobile ? 14 : 15,
889. ),
890. hintStyle: GoogleFonts.poppins(
891. color: Colors.grey[400],
892. fontSize: isMobile ? 14 : 15,
893. ),
894. ),
895. minLines: 4,
896. maxLines: 6,
897. validator: (value) {
898. if (value == null || value.isEmpty) {
899. return 'Please provide your feedback.';
900. }
901. return null;
902. },
903. ),
904. ),
905. SizedBox(height: 24),
906. AnimatedContainer(
907. duration: Duration(milliseconds: 300),
908. width: double.infinity,
909. child: ElevatedButton(
910. onPressed: \_isSubmitting ? null : \_submitFeedback,
911. style: ElevatedButton.styleFrom(
912. backgroundColor: Colors.indigo[700],
913. foregroundColor: Colors.white,
914. padding: EdgeInsets.symmetric(
915. vertical: isMobile ? 14 : 16,
916. horizontal: isMobile ? 24 : 32,
917. ),
918. shape: RoundedRectangleBorder(
919. borderRadius: BorderRadius.circular(12),
920. ),
921. elevation: 2,
922. shadowColor: Colors.indigo.withOpacity(0.3),
923. ),
924. child: \_isSubmitting
925. ? SizedBox(
926. height: isMobile ? 20 : 24,
927. width: isMobile ? 20 : 24,
928. child: CircularProgressIndicator(
929. strokeWidth: 2,
930. valueColor: AlwaysStoppedAnimation<Color>(Colors.white),
931. ),
932. )
933. : Text(
934. 'Submit Feedback',
935. style: GoogleFonts.poppins(
936. fontSize: isMobile ? 15 : 16,
937. fontWeight: FontWeight.w600,
938. ),
939. ),
940. ),
941. ),
942. ],
943. ),
944. );
945. }
946. }
947. class HomePage extends StatelessWidget {
948. @override
949. Widget build(BuildContext context) {
950. return SingleChildScrollView(
951. child: Column(
952. children: [
953. HeroSection(),
954. MissionVisionSection(),
955. ServicesSection(),
956. ContactSection(),
957. FooterSection(),
958. ],
959. ),
960. );
961. }
962. }
963. class AppointmentPage extends StatelessWidget {
964. @override
965. Widget build(BuildContext context) {
966. return SingleChildScrollView(
967. child: Column(children: [AppointmentSection(), FooterSection()]),
968. );
969. }
970. }
971. class DashboardPage extends StatelessWidget {
972. @override
973. Widget build(BuildContext context) {
974. return SingleChildScrollView(
975. child: Column(children: [DashboardSection(), FooterSection()]),
976. );
977. }
978. }
979. class AuthService extends ChangeNotifier {
980. bool \_isLoggedIn = false;
981. String? \_username;
982. String? \_email;
983. static const String \_usersKey = 'users';
984. bool get isLoggedIn => \_isLoggedIn;
985. String? get username => \_username;
986. String? get email => \_email;
987. Future<void> loadAuthState() async {
988. final prefs = await SharedPreferences.getInstance();
989. \_isLoggedIn = prefs.getBool('isLoggedIn') ?? false;
990. \_username = prefs.getString('username');
991. \_email = prefs.getString('email');
992. notifyListeners();
993. }
994. Future<List<Map<String, dynamic>>> \_loadUsers() async {
995. final prefs = await SharedPreferences.getInstance();
996. final usersJson = prefs.getString(\_usersKey);
997. if (usersJson == null) return [];
998. return List<Map<String, dynamic>>.from(jsonDecode(usersJson));
999. }
1000. Future<void> \_saveUsers(List<Map<String, dynamic>> users) async {
1001. final prefs = await SharedPreferences.getInstance();
1002. await prefs.setString(\_usersKey, jsonEncode(users));
1003. }
1004. Future<void> login(String email, String password) async {
1005. final prefs = await SharedPreferences.getInstance();
1006. final users = await \_loadUsers();
1007. final user = users.firstWhere(
1008. (u) => u['email'] == email && u['password'] == password,
1009. orElse: () => {},
1010. );
1011. if (user.isNotEmpty) {
1012. \_isLoggedIn = true;
1013. \_username = user['username'];
1014. \_email = email;
1015. await prefs.setBool('isLoggedIn', true);
1016. await prefs.setString('username', user['username']);
1017. await prefs.setString('email', email);
1018. notifyListeners();
1019. } else {
1020. throw Exception('Invalid credentials');
1021. }
1022. }
1023. Future<void> signup(String username, String email, String password) async {
1024. final prefs = await SharedPreferences.getInstance();
1025. final users = await \_loadUsers();
1026. if (users.any((u) => u['username'] == username)) {
1027. throw Exception('Username already exists');
1028. }
1029. users.add({'username': username, 'email': email, 'password': password});
1030. await \_saveUsers(users);
1031. \_isLoggedIn = true;
1032. \_username = username;
1033. \_email = email;
1034. await prefs.setBool('isLoggedIn', true);
1035. await prefs.setString('username', username);
1036. await prefs.setString('email', email);
1037. notifyListeners();
1038. }
1039. Future<void> logout() async {
1040. final prefs = await SharedPreferences.getInstance();
1041. \_isLoggedIn = false;
1042. \_username = null;
1043. \_email = null;
1044. await prefs.setBool('isLoggedIn', false);
1045. await prefs.remove('username');
1046. await prefs.remove('email');
1047. notifyListeners();
1048. }
1049. }
1050. class AppointmentService extends ChangeNotifier {
1051. static const String \_appointmentsKey = 'appointments';
1052. Future<List<Map<String, dynamic>>> loadAppointments() async {
1053. final prefs = await SharedPreferences.getInstance();
1054. final appointmentsJson = prefs.getString(\_appointmentsKey);
1055. if (appointmentsJson == null) return [];
1056. return List<Map<String, dynamic>>.from(jsonDecode(appointmentsJson));
1057. }
1058. Future<void> saveAppointment(Map<String, dynamic> appointment) async {
1059. final prefs = await SharedPreferences.getInstance();
1060. final appointments = await loadAppointments();
1061. appointments.add(appointment);
1062. await prefs.setString(\_appointmentsKey, jsonEncode(appointments));
1063. notifyListeners();
1064. }
1065. Future<void> clearAppointments() async {
1066. final prefs = await SharedPreferences.getInstance();
1067. await prefs.remove(\_appointmentsKey);
1068. notifyListeners();
1069. }
1070. }
1071. class FeedbackService extends ChangeNotifier {
1072. static const String \_feedbackKey = 'feedback';
1073. Future<List<Map<String, dynamic>>> loadFeedback() async {
1074. final prefs = await SharedPreferences.getInstance();
1075. final feedbackJson = prefs.getString(\_feedbackKey);
1076. if (feedbackJson == null) return [];
1077. return List<Map<String, dynamic>>.from(jsonDecode(feedbackJson));
1078. }
1079. Future<void> saveFeedback(Map<String, dynamic> feedback) async {
1080. final prefs = await SharedPreferences.getInstance();
1081. final feedbackList = await loadFeedback();
1082. feedbackList.add(feedback);
1083. await prefs.setString(\_feedbackKey, jsonEncode(feedbackList));
1084. notifyListeners();
1085. }
1086. Future<void> clearFeedback() async {
1087. final prefs = await SharedPreferences.getInstance();
1088. await prefs.remove(\_feedbackKey);
1089. notifyListeners();
1090. }
1091. }
1092. class AuthDialog extends StatefulWidget {
1093. const AuthDialog({Key? key}) : super(key: key);
1094. @override
1095. State<AuthDialog> createState() => \_AuthDialogState();
1096. }
1097. class \_AuthDialogState extends State<AuthDialog>
1098. with SingleTickerProviderStateMixin {
1099. late TabController \_tabController;
1100. final \_loginFormKey = GlobalKey<FormState>();
1101. final \_signupFormKey = GlobalKey<FormState>();
1102. final \_loginEmailController = TextEditingController();
1103. final \_loginPasswordController = TextEditingController();
1104. final \_signupUsernameController = TextEditingController();
1105. final \_signupEmailController = TextEditingController();
1106. final \_signupPasswordController = TextEditingController();
1107. final \_confirmPasswordController = TextEditingController();
1108. @override
1109. void initState() {
1110. super.initState();
1111. \_tabController = TabController(length: 2, vsync: this);
1112. }
1113. @override
1114. void dispose() {
1115. \_tabController.dispose();
1116. \_loginEmailController.dispose();
1117. \_loginPasswordController.dispose();
1118. \_signupUsernameController.dispose();
1119. \_signupEmailController.dispose();
1120. \_signupPasswordController.dispose();
1121. \_confirmPasswordController.dispose();
1122. super.dispose();
1123. }
1124. void \_submitLogin() async {
1125. if (\_loginFormKey.currentState!.validate()) {
1126. try {
1127. await Provider.of<AuthService>(
1128. context,
1129. listen: false,
1130. ).login(\_loginEmailController.text, \_loginPasswordController.text);
1131. Navigator.pop(context);
1132. ScaffoldMessenger.of(context).showSnackBar(
1133. const SnackBar(
1134. content: Text('Login successful!'),
1135. backgroundColor: Colors.green,
1136. ),
1137. );
1138. } catch (e) {
1139. ScaffoldMessenger.of(context).showSnackBar(
1140. SnackBar(
1141. content: Text('Login failed: $e'),
1142. backgroundColor: Colors.red,
1143. ),
1144. );
1145. }
1146. }
1147. }
1148. void \_submitSignup() async {
1149. if (\_signupFormKey.currentState!.validate()) {
1150. if (\_signupPasswordController.text != \_confirmPasswordController.text) {
1151. ScaffoldMessenger.of(context).showSnackBar(
1152. const SnackBar(
1153. content: Text('Passwords do not match'),
1154. backgroundColor: Colors.red,
1155. ),
1156. );
1157. return;
1158. }
1159. try {
1160. await Provider.of<AuthService>(context, listen: false).signup(
1161. \_signupUsernameController.text,
1162. \_signupEmailController.text,
1163. \_signupPasswordController.text,
1164. );
1165. Navigator.pop(context);
1166. ScaffoldMessenger.of(context).showSnackBar(
1167. const SnackBar(
1168. content: Text('Sign up successful!'),
1169. backgroundColor: Colors.green,
1170. ),
1171. );
1172. } catch (e) {
1173. ScaffoldMessenger.of(context).showSnackBar(
1174. SnackBar(
1175. content: Text('Sign up failed: $e'),
1176. backgroundColor: Colors.red,
1177. ),
1178. );
1179. }
1180. }
1181. }
1182. @override
1183. Widget build(BuildContext context) {
1184. final isMobile = MediaQuery.of(context).size.width < 600;
1185. final theme = Theme.of(context);
1186. return Dialog(
1187. shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(16)),
1188. insetPadding: EdgeInsets.symmetric(
1189. horizontal: isMobile ? 16 : 24,
1190. vertical: 24,
1191. ),
1192. child: ConstrainedBox(
1193. constraints: BoxConstraints(
1194. maxWidth: isMobile ? 400 : 500,
1195. minWidth: isMobile ? 300 : 400,
1196. ),
1197. child: SingleChildScrollView(
1198. child: Padding(
1199. padding: EdgeInsets.all(isMobile ? 16 : 24),
1200. child: Column(
1201. mainAxisSize: MainAxisSize.min,
1202. children: [
1203. Row(
1204. mainAxisAlignment: MainAxisAlignment.center,
1205. children: [
1206. Icon(
1207. Icons.medical\_services,
1208. color: theme.primaryColor,
1209. size: isMobile ? 24 : 32,
1210. ),
1211. SizedBox(width: 8),
1212. Text(
1213. 'Global Care',
1214. style: theme.textTheme.titleLarge?.copyWith(
1215. color: theme.primaryColor,
1216. fontWeight: FontWeight.bold,
1217. ),
1218. ),
1219. ],
1220. ),
1221. SizedBox(height: 16),
1222. Container(
1223. decoration: BoxDecoration(
1224. color: theme.colorScheme.surfaceVariant.withOpacity(0.5),
1225. borderRadius: BorderRadius.circular(12),
1226. ),
1227. child: ListenableBuilder(
1228. listenable: \_tabController.animation!,
1229. builder: (context, \_) {
1230. final index = \_tabController.animation!.value.round();
1231. return TabBar(
1232. controller: \_tabController,
1233. labelColor: Colors.white,
1234. unselectedLabelColor: theme.primaryColor,
1235. indicator: BoxDecoration(
1236. color: theme.primaryColor,
1237. borderRadius: index == 0
1238. ? BorderRadius.only(
1239. topLeft: Radius.circular(12),
1240. bottomLeft: Radius.circular(12),
1241. )
1242. : BorderRadius.only(
1243. topRight: Radius.circular(12),
1244. bottomRight: Radius.circular(12),
1245. ),
1246. ),
1247. indicatorSize: TabBarIndicatorSize.tab,
1248. dividerColor: Colors.transparent,
1249. tabs: [
1250. Tab(
1251. child: Row(
1252. mainAxisAlignment: MainAxisAlignment.center,
1253. children: [
1254. Icon(Icons.login, size: isMobile ? 18 : 20),
1255. SizedBox(width: 4),
1256. Text(
1257. 'Login',
1258. style: GoogleFonts.poppins(
1259. fontSize: isMobile ? 14 : 16,
1260. fontWeight: FontWeight.w500,
1261. ),
1262. ),
1263. ],
1264. ),
1265. ),
1266. Tab(
1267. child: Row(
1268. mainAxisAlignment: MainAxisAlignment.center,
1269. children: [
1270. Icon(
1271. Icons.person\_add,
1272. size: isMobile ? 18 : 20,
1273. ),
1274. SizedBox(width: 4),
1275. Text(
1276. 'Sign Up',
1277. style: GoogleFonts.poppins(
1278. fontSize: isMobile ? 14 : 16,
1279. fontWeight: FontWeight.w500,
1280. ),
1281. ),
1282. ],
1283. ),
1284. ),
1285. ],
1286. );
1287. },
1288. ),
1289. ),
1290. SizedBox(height: 16),
1291. SizedBox(
1292. height: isMobile ? 400 : 450,
1293. child: TabBarView(
1294. controller: \_tabController,
1295. children: [
1296. // Login Tab
1297. Form(
1298. key: \_loginFormKey,
1299. child: Column(
1300. mainAxisSize: MainAxisSize.min,
1301. children: [
1302. TextFormField(
1303. controller: \_loginEmailController,
1304. decoration: InputDecoration(
1305. labelText: 'Email',
1306. prefixIcon: Icon(Icons.email),
1307. border: OutlineInputBorder(
1308. borderRadius: BorderRadius.circular(12),
1309. ),
1310. filled: true,
1311. fillColor: Colors.indigo.withOpacity(0.05),
1312. ),
1313. validator: (value) {
1314. if (value == null ||
1315. value.isEmpty ||
1316. !value.contains('@')) {
1317. return 'Please enter a valid email';
1318. }
1319. return null;
1320. },
1321. ),
1322. SizedBox(height: 16),
1323. TextFormField(
1324. controller: \_loginPasswordController,
1325. decoration: InputDecoration(
1326. labelText: 'Password',
1327. prefixIcon: Icon(Icons.lock),
1328. border: OutlineInputBorder(
1329. borderRadius: BorderRadius.circular(12),
1330. ),
1331. filled: true,
1332. fillColor: Colors.indigo.withOpacity(0.05),
1333. ),
1334. obscureText: true,
1335. validator: (value) {
1336. if (value == null || value.isEmpty) {
1337. return 'Please enter password';
1338. }
1339. return null;
1340. },
1341. ),
1342. SizedBox(height: 16),
1343. Align(
1344. alignment: Alignment.centerRight,
1345. child: TextButton(
1346. onPressed: () {
1347. // Forgot password
1348. },
1349. child: Text(
1350. 'Forgot Password?',
1351. style: GoogleFonts.poppins(
1352. color: theme.primaryColor,
1353. fontSize: isMobile ? 12 : 14,
1354. ),
1355. ),
1356. ),
1357. ),
1358. SizedBox(height: 16),
1359. ElevatedButton(
1360. onPressed: \_submitLogin,
1361. style: ElevatedButton.styleFrom(
1362. backgroundColor: theme.primaryColor,
1363. foregroundColor: Colors.white,
1364. minimumSize: Size(double.infinity, 50),
1365. shape: RoundedRectangleBorder(
1366. borderRadius: BorderRadius.circular(12),
1367. ),
1368. ),
1369. child: Text(
1370. 'Login',
1371. style: GoogleFonts.poppins(
1372. fontSize: isMobile ? 15 : 16,
1373. fontWeight: FontWeight.w600,
1374. ),
1375. ),
1376. ),
1377. SizedBox(height: 16),
1378. Row(
1379. mainAxisAlignment: MainAxisAlignment.center,
1380. children: [
1381. Text(
1382. "Don't have an account?",
1383. style: GoogleFonts.poppins(
1384. fontSize: isMobile ? 12 : 14,
1385. color: Colors.grey[600],
1386. ),
1387. ),
1388. TextButton(
1389. onPressed: () => \_tabController.animateTo(1),
1390. child: Text(
1391. 'Sign Up',
1392. style: GoogleFonts.poppins(
1393. color: theme.primaryColor,
1394. fontSize: isMobile ? 12 : 14,
1395. fontWeight: FontWeight.w600,
1396. ),
1397. ),
1398. ),
1399. ],
1400. ),
1401. ],
1402. ),
1403. ),
1404. // Sign Up Tab
1405. Form(
1406. key: \_signupFormKey,
1407. child: Column(
1408. mainAxisSize: MainAxisSize.min,
1409. children: [
1410. TextFormField(
1411. controller: \_signupUsernameController,
1412. decoration: InputDecoration(
1413. labelText: 'Username',
1414. prefixIcon: Icon(Icons.person),
1415. border: OutlineInputBorder(
1416. borderRadius: BorderRadius.circular(12),
1417. ),
1418. filled: true,
1419. fillColor: Colors.indigo.withOpacity(0.05),
1420. ),
1421. validator: (value) {
1422. if (value == null || value.isEmpty) {
1423. return 'Please enter username';
1424. }
1425. return null;
1426. },
1427. ),
1428. SizedBox(height: 16),
1429. TextFormField(
1430. controller: \_signupEmailController,
1431. decoration: InputDecoration(
1432. labelText: 'Email',
1433. prefixIcon: Icon(Icons.email),
1434. border: OutlineInputBorder(
1435. borderRadius: BorderRadius.circular(12),
1436. ),
1437. filled: true,
1438. fillColor: Colors.indigo.withOpacity(0.05),
1439. ),
1440. validator: (value) {
1441. if (value == null ||
1442. value.isEmpty ||
1443. !value.contains('@')) {
1444. return 'Please enter valid email';
1445. }
1446. return null;
1447. },
1448. ),
1449. SizedBox(height: 16),
1450. TextFormField(
1451. controller: \_signupPasswordController,
1452. decoration: InputDecoration(
1453. labelText: 'Password',
1454. prefixIcon: Icon(Icons.lock),
1455. helperText: 'At least 6 characters',
1456. border: OutlineInputBorder(
1457. borderRadius: BorderRadius.circular(12),
1458. ),
1459. filled: true,
1460. fillColor: Colors.indigo.withOpacity(0.05),
1461. ),
1462. obscureText: true,
1463. validator: (value) {
1464. if (value == null ||
1465. value.isEmpty ||
1466. value.length < 6) {
1467. return 'Password must be at least 6 characters';
1468. }
1469. return null;
1470. },
1471. ),
1472. SizedBox(height: 16),
1473. TextFormField(
1474. controller: \_confirmPasswordController,
1475. decoration: InputDecoration(
1476. labelText: 'Confirm Password',
1477. prefixIcon: Icon(Icons.lock),
1478. border: OutlineInputBorder(
1479. borderRadius: BorderRadius.circular(12),
1480. ),
1481. filled: true,
1482. fillColor: Colors.indigo.withOpacity(0.05),
1483. ),
1484. obscureText: true,
1485. validator: (value) {
1486. if (value == null || value.isEmpty) {
1487. return 'Please confirm password';
1488. }
1489. return null;
1490. },
1491. ),
1492. SizedBox(height: 16),
1493. ElevatedButton(
1494. onPressed: \_submitSignup,
1495. style: ElevatedButton.styleFrom(
1496. backgroundColor: theme.primaryColor,
1497. foregroundColor: Colors.white,
1498. minimumSize: Size(double.infinity, 50),
1499. shape: RoundedRectangleBorder(
1500. borderRadius: BorderRadius.circular(12),
1501. ),
1502. ),
1503. child: Text(
1504. 'Create Account',
1505. style: GoogleFonts.poppins(
1506. fontSize: isMobile ? 15 : 16,
1507. fontWeight: FontWeight.w600,
1508. ),
1509. ),
1510. ),
1511. SizedBox(height: 16),
1512. Row(
1513. mainAxisAlignment: MainAxisAlignment.center,
1514. children: [
1515. Text(
1516. 'Already have an account?',
1517. style: GoogleFonts.poppins(
1518. fontSize: isMobile ? 12 : 14,
1519. color: Colors.grey[600],
1520. ),
1521. ),
1522. TextButton(
1523. onPressed: () => \_tabController.animateTo(0),
1524. child: Text(
1525. 'Login',
1526. style: GoogleFonts.poppins(
1527. color: theme.primaryColor,
1528. fontSize: isMobile ? 12 : 14,
1529. fontWeight: FontWeight.w600,
1530. ),
1531. ),
1532. ),
1533. ],
1534. ),
1535. ],
1536. ),
1537. ),
1538. ],
1539. ),
1540. ),
1541. ],
1542. ),
1543. ),
1544. ),
1545. ),
1546. );
1547. }
1548. }
1549. class HeroSection extends StatelessWidget {
1550. const HeroSection({Key? key}) : super(key: key);
1551. @override
1552. Widget build(BuildContext context) {
1553. final authService = Provider.of<AuthService>(context);
1554. return LayoutBuilder(
1555. builder: (context, constraints) {
1556. final isMobile = constraints.maxWidth < 600;
1557. final screenHeight = MediaQuery.of(context).size.height;
1558. return Stack(
1559. children: [
1560. Container(
1561. height: screenHeight,
1562. width: double.infinity,
1563. decoration: BoxDecoration(
1564. image: DecorationImage(
1565. image: NetworkImage(
1566. 'https://upload.wikimedia.org/wikipedia/commons/thumb/e/e1/2094Calamba\_City\_Canlubang\_Roads\_Landmarks\_Barangays\_31.jpg/1200px-2094Calamba\_City\_Canlubang\_Roads\_Landmarks\_Barangays\_31.jpg',
1567. ),
1568. fit: BoxFit.cover,
1569. ),
1570. ),
1571. ),
1572. Container(
1573. height: screenHeight,
1574. width: double.infinity,
1575. decoration: BoxDecoration(
1576. gradient: LinearGradient(
1577. colors: [Color(0xAA1E40AF), Color(0x803B82F6)],
1578. begin: Alignment.topLeft,
1579. end: Alignment.bottomRight,
1580. ),
1581. ),
1582. ),
1583. Container(
1584. height: screenHeight,
1585. padding: EdgeInsets.symmetric(
1586. horizontal: constraints.maxWidth \* 0.05,
1587. vertical: isMobile ? 16 : 24,
1588. ),
1589. alignment: Alignment.center,
1590. child: Column(
1591. mainAxisAlignment: MainAxisAlignment.center,
1592. children: [
1593. AnimatedOpacity(
1594. opacity: 1.0,
1595. duration: Duration(milliseconds: 1000),
1596. child: Text(
1597. 'Welcome to Global Care Medical Center',
1598. textAlign: TextAlign.center,
1599. style: GoogleFonts.poppins(
1600. color: Colors.white,
1601. fontSize: isMobile ? 24 : 36,
1602. fontWeight: FontWeight.w700,
1603. shadows: [
1604. Shadow(
1605. color: Colors.black26,
1606. blurRadius: 4,
1607. offset: Offset(2, 2),
1608. ),
1609. ],
1610. ),
1611. ),
1612. ),
1613. SizedBox(height: isMobile ? 12 : 16),
1614. AnimatedSlide(
1615. offset: Offset(0, 0),
1616. duration: Duration(milliseconds: 1200),
1617. curve: Curves.easeOut,
1618. child: Text(
1619. 'Providing compassionate care and cutting-edge medical services to our community. At Global, We Care!',
1620. textAlign: TextAlign.center,
1621. style: GoogleFonts.poppins(
1622. color: Colors.white.withOpacity(0.95),
1623. fontSize: isMobile ? 14 : 18,
1624. fontWeight: FontWeight.w400,
1625. ),
1626. ),
1627. ),
1628. SizedBox(height: isMobile ? 12 : 24),
1629. Wrap(
1630. alignment: WrapAlignment.center,
1631. spacing: 12,
1632. runSpacing: 8,
1633. children: [
1634. ElevatedButton(
1635. style: ElevatedButton.styleFrom(
1636. backgroundColor: Colors.white,
1637. foregroundColor: Colors.indigo[700],
1638. padding: EdgeInsets.symmetric(
1639. horizontal: isMobile ? 20 : 28,
1640. vertical: isMobile ? 12 : 16,
1641. ),
1642. shape: RoundedRectangleBorder(
1643. borderRadius: BorderRadius.circular(24),
1644. ),
1645. elevation: 2,
1646. shadowColor: Colors.black.withOpacity(0.2),
1647. ),
1648. onPressed: () {
1649. if (authService.isLoggedIn) {
1650. try {
1651. final tabController = DefaultTabController.of(
1652. context,
1653. );
1654. if (tabController != null) {
1655. tabController.animateTo(1);
1656. (context as Element).markNeedsBuild();
1657. } else {
1658. ScaffoldMessenger.of(context).showSnackBar(
1659. SnackBar(
1660. content: Text(
1661. 'Navigation error. Please try again.',
1662. ),
1663. ),
1664. );
1665. }
1666. } catch (e) {
1667. ScaffoldMessenger.of(context).showSnackBar(
1668. SnackBar(content: Text('Navigation error: $e')),
1669. );
1670. }
1671. } else {
1672. showDialog(
1673. context: context,
1674. builder: (context) => AuthDialog(),
1675. );
1676. }
1677. },
1678. child: Text(
1679. 'Book Appointment',
1680. style: GoogleFonts.poppins(
1681. color: Colors.indigo[700],
1682. fontSize: isMobile ? 14 : 16,
1683. fontWeight: FontWeight.w600,
1684. ),
1685. ),
1686. ),
1687. OutlinedButton(
1688. style: OutlinedButton.styleFrom(
1689. backgroundColor: Colors.transparent,
1690. foregroundColor: Colors.white,
1691. padding: EdgeInsets.symmetric(
1692. horizontal: isMobile ? 20 : 28,
1693. vertical: isMobile ? 12 : 16,
1694. ),
1695. side: BorderSide(color: Colors.white, width: 1.5),
1696. shape: RoundedRectangleBorder(
1697. borderRadius: BorderRadius.circular(24),
1698. ),
1699. ),
1700. onPressed: () {
1701. ScaffoldMessenger.of(context).showSnackBar(
1702. SnackBar(content: Text('Learn More clicked!')),
1703. );
1704. },
1705. child: Text(
1706. 'Learn More',
1707. style: GoogleFonts.poppins(
1708. color: Colors.white,
1709. fontSize: isMobile ? 14 : 16,
1710. fontWeight: FontWeight.w500,
1711. ),
1712. ),
1713. ),
1714. ],
1715. ),
1716. SizedBox(height: isMobile ? 12 : 24),
1717. HeroStats(isMobile: isMobile),
1718. ],
1719. ),
1720. ),
1721. ],
1722. );
1723. },
1724. );
1725. }
1726. }
1727. class HeroStats extends StatelessWidget {
1728. final bool isMobile;
1729. HeroStats({required this.isMobile});
1730. @override
1731. Widget build(BuildContext context) {
1732. return AnimatedOpacity(
1733. opacity: 1.0,
1734. duration: Duration(milliseconds: 1500),
1735. child: Container(
1736. margin: EdgeInsets.only(top: isMobile ? 12 : 24),
1737. padding: EdgeInsets.symmetric(
1738. horizontal: isMobile ? 12 : 24,
1739. vertical: 12,
1740. ),
1741. decoration: BoxDecoration(
1742. color: Colors.white.withOpacity(0.15),
1743. borderRadius: BorderRadius.circular(16),
1744. border: Border.all(color: Colors.white.withOpacity(0.2)),
1745. boxShadow: [BoxShadow(color: Colors.black12, blurRadius: 6)],
1746. ),
1747. child: Wrap(
1748. alignment: WrapAlignment.center,
1749. spacing: isMobile ? 20 : 40,
1750. runSpacing: 12,
1751. children: [
1752. StatItem(
1753. icon: Icons.medical\_services,
1754. number: '210+',
1755. label: 'Expert Doctors',
1756. isMobile: isMobile,
1757. ),
1758. StatItem(
1759. icon: Icons.local\_hospital,
1760. number: '150+',
1761. label: 'Departments',
1762. isMobile: isMobile,
1763. ),
1764. StatItem(
1765. icon: Icons.people,
1766. number: '10K+',
1767. label: 'Happy Patients',
1768. isMobile: isMobile,
1769. ),
1770. ],
1771. ),
1772. ),
1773. );
1774. }
1775. }
1776. class StatItem extends StatelessWidget {
1777. final IconData icon;
1778. final String number;
1779. final String label;
1780. final bool isMobile;
1781. const StatItem({
1782. required this.icon,
1783. required this.number,
1784. required this.label,
1785. required this.isMobile,
1786. });
1787. @override
1788. Widget build(BuildContext context) {
1789. return Column(
1790. children: [
1791. Icon(icon, color: Colors.amber, size: isMobile ? 24 : 32),
1792. SizedBox(height: 4),
1793. Text(
1794. number,
1795. style: GoogleFonts.poppins(
1796. color: Colors.white,
1797. fontSize: isMobile ? 16 : 20,
1798. fontWeight: FontWeight.w600,
1799. ),
1800. ),
1801. Text(
1802. label,
1803. style: GoogleFonts.poppins(
1804. color: Colors.white.withOpacity(0.9),
1805. fontSize: isMobile ? 12 : 14,
1806. fontWeight: FontWeight.w400,
1807. ),
1808. ),
1809. ],
1810. );
1811. }
1812. }
1813. class MissionVisionSection extends StatelessWidget {
1814. @override
1815. Widget build(BuildContext context) {
1816. return LayoutBuilder(
1817. builder: (context, constraints) {
1818. final isMobile = constraints.maxWidth < 600;
1819. return Container(
1820. color: Color(0xFFF8FAFC),
1821. padding: EdgeInsets.symmetric(
1822. vertical: isMobile ? 24 : 40,
1823. horizontal: constraints.maxWidth \* 0.05,
1824. ),
1825. child: Column(
1826. children: [
1827. SectionHeader(title: 'Our Mission & Vision'),
1828. SizedBox(height: isMobile ? 12 : 24),
1829. isMobile
1830. ? Column(
1831. children: [
1832. MissionVisionCard(
1833. title: 'Our Mission',
1834. icon: Icons.track\_changes,
1835. content:
1836. "To improve patient's lives by providing safe, effective and suitable medical services performed only by high capable staffs and medical professionals while establishing an environment where customers, employees and stakeholders are valued and involved in continuously improving the quality of our services towards a healthier community.",
1837. isMobile: isMobile,
1838. ),
1839. SizedBox(height: 16),
1840. MissionVisionCard(
1841. title: 'Our Vision',
1842. icon: Icons.remove\_red\_eye,
1843. content:
1844. "Global Care Medical Center of Canlubang is the optimal healthcare provider in the Community that delivers excellent medical service in the most professional and compassionate way at the most reasonable cost.",
1845. isMobile: isMobile,
1846. ),
1847. ],
1848. )
1849. : Row(
1850. crossAxisAlignment: CrossAxisAlignment.start,
1851. children: [
1852. Expanded(
1853. child: MissionVisionCard(
1854. title: 'Our Mission',
1855. icon: Icons.track\_changes,
1856. content:
1857. "To improve patient's lives by providing safe, effective and suitable medical services performed only by high capable staffs and medical professionals while establishing an environment where customers, employees and stakeholders are valued and involved in continuously improving the quality of our services towards a healthier community.",
1858. isMobile: isMobile,
1859. ),
1860. ),
1861. SizedBox(width: 24),
1862. Expanded(
1863. child: MissionVisionCard(
1864. title: 'Our Vision',
1865. icon: Icons.remove\_red\_eye,
1866. content:
1867. "Global Care Medical Center of Canlubang is the optimal healthcare provider in the Community that delivers excellent medical service in the most professional and compassionate way at the most reasonable cost.",
1868. isMobile: isMobile,
1869. ),
1870. ),
1871. ],
1872. ),
1873. ],
1874. ),
1875. );
1876. },
1877. );
1878. }
1879. }
1880. class MissionVisionCard extends StatelessWidget {
1881. final String title;
1882. final IconData icon;
1883. final String content;
1884. final bool isMobile;
1885. MissionVisionCard({
1886. required this.title,
1887. required this.icon,
1888. required this.content,
1889. required this.isMobile,
1890. });
1891. @override
1892. Widget build(BuildContext context) {
1893. return AnimatedOpacity(
1894. opacity: 1.0,
1895. duration: Duration(milliseconds: 800),
1896. child: Card(
1897. child: Padding(
1898. padding: EdgeInsets.all(isMobile ? 16 : 24),
1899. child: Column(
1900. children: [
1901. CircleAvatar(
1902. backgroundColor: Colors.indigo.withOpacity(0.1),
1903. child: Icon(icon, color: Colors.indigo[700]),
1904. radius: isMobile ? 24 : 28,
1905. ),
1906. SizedBox(height: 12),
1907. Text(
1908. title,
1909. style: GoogleFonts.poppins(
1910. fontSize: isMobile ? 16 : 18,
1911. fontWeight: FontWeight.w500,
1912. ),
1913. ),
1914. SizedBox(height: 8),
1915. Text(
1916. content,
1917. style: GoogleFonts.poppins(
1918. fontSize: isMobile ? 14 : 16,
1919. fontWeight: FontWeight.w400,
1920. ),
1921. textAlign: TextAlign.center,
1922. ),
1923. ],
1924. ),
1925. ),
1926. ),
1927. );
1928. }
1929. }
1930. class ServicesSection extends StatelessWidget {
1931. final List<Map<String, dynamic>> services = [
1932. {
1933. 'icon': Icons.local\_hospital,
1934. 'title': 'Emergency Care',
1935. 'desc':
1936. 'Available 24/7 for all emergency medical needs with state-of-the-art facilities.',
1937. },
1938. {
1939. 'icon': Icons.favorite,
1940. 'title': 'Primary Care',
1941. 'desc':
1942. 'Comprehensive primary healthcare services for patients of all ages.',
1943. },
1944. {
1945. 'icon': Icons.healing,
1946. 'title': 'Specialized Treatment',
1947. 'desc':
1948. 'Expert specialists providing advanced care across multiple medical disciplines.',
1949. },
1950. {
1951. 'icon': Icons.science,
1952. 'title': 'Laboratory Services',
1953. 'desc': 'Advanced diagnostic testing with quick and accurate results.',
1954. },
1955. {
1956. 'icon': Icons.medication,
1957. 'title': 'Pharmacy',
1958. 'desc':
1959. 'On-site pharmacy providing prescription medications and health products.',
1960. },
1961. {
1962. 'icon': Icons.fitness\_center,
1963. 'title': 'Rehabilitation',
1964. 'desc':
1965. 'Comprehensive rehabilitation services to help patients recover and regain independence.',
1966. },
1967. ];
1968. @override
1969. Widget build(BuildContext context) {
1970. return LayoutBuilder(
1971. builder: (context, constraints) {
1972. final isMobile = constraints.maxWidth < 600;
1973. final crossAxisCount = constraints.maxWidth < 600
1974. ? 1
1975. : constraints.maxWidth < 900
1976. ? 2
1977. : 3;
1978. return Container(
1979. color: Colors.white,
1980. padding: EdgeInsets.symmetric(
1981. vertical: isMobile ? 24 : 40,
1982. horizontal: constraints.maxWidth \* 0.05,
1983. ),
1984. child: Column(
1985. children: [
1986. SectionHeader(
1987. title: 'Our Services',
1988. subtitle:
1989. 'Comprehensive medical services to meet all your healthcare needs.',
1990. ),
1991. SizedBox(height: isMobile ? 12 : 24),
1992. GridView.count(
1993. shrinkWrap: true,
1994. physics: NeverScrollableScrollPhysics(),
1995. crossAxisCount: crossAxisCount,
1996. crossAxisSpacing: isMobile ? 16 : 24,
1997. mainAxisSpacing: isMobile ? 16 : 24,
1998. childAspectRatio: isMobile ? 1.3 : 1.1,
1999. children: services
2000. .asMap()
2001. .entries
2002. .map(
2003. (entry) => AnimatedOpacity(
2004. opacity: 1.0,
2005. duration: Duration(milliseconds: 500 + entry.key \* 200),
2006. child: Card(
2007. child: Padding(
2008. padding: EdgeInsets.all(isMobile ? 16 : 20),
2009. child: Column(
2010. children: [
2011. CircleAvatar(
2012. backgroundColor: Colors.indigo.withOpacity(
2013. 0.1,
2014. ),
2015. radius: isMobile ? 28 : 36,
2016. child: Icon(
2017. entry.value['icon'],
2018. color: Colors.indigo[700],
2019. size: isMobile ? 24 : 28,
2020. ),
2021. ),
2022. SizedBox(height: 12),
2023. Text(
2024. entry.value['title'],
2025. style: GoogleFonts.poppins(
2026. fontSize: isMobile ? 16 : 18,
2027. fontWeight: FontWeight.w500,
2028. ),
2029. textAlign: TextAlign.center,
2030. ),
2031. SizedBox(height: 8),
2032. Text(
2033. entry.value['desc'],
2034. style: GoogleFonts.poppins(
2035. fontSize: isMobile ? 14 : 16,
2036. fontWeight: FontWeight.w400,
2037. ),
2038. textAlign: TextAlign.center,
2039. ),
2040. Spacer(),
2041. OutlinedButton.icon(
2042. onPressed: () {
2043. ScaffoldMessenger.of(context).showSnackBar(
2044. SnackBar(
2045. content: Text(
2046. '${entry.value['title']} Learn More clicked!',
2047. ),
2048. ),
2049. );
2050. },
2051. icon: Icon(
2052. Icons.arrow\_right\_alt,
2053. color: Colors.indigo[700],
2054. size: isMobile ? 20 : 24,
2055. ),
2056. label: Text('Learn More'),
2057. ),
2058. ],
2059. ),
2060. ),
2061. ),
2062. ),
2063. )
2064. .toList(),
2065. ),
2066. ],
2067. ),
2068. );
2069. },
2070. );
2071. }
2072. }
2073. class AppointmentSection extends StatefulWidget {
2074. const AppointmentSection({Key? key}) : super(key: key);
2075. @override
2076. State<AppointmentSection> createState() => \_AppointmentSectionState();
2077. }
2078. class \_AppointmentSectionState extends State<AppointmentSection> {
2079. int tabIndex = 0;
2080. @override
2081. Widget build(BuildContext context) {
2082. return Consumer<AuthService>(
2083. builder: (context, authService, child) {
2084. return LayoutBuilder(
2085. builder: (context, constraints) {
2086. final isMobile = constraints.maxWidth < 600;
2087. return Container(
2088. color: Color(0xFFF8FAFC),
2089. padding: EdgeInsets.symmetric(
2090. vertical: isMobile ? 24 : 40,
2091. horizontal: constraints.maxWidth \* 0.05,
2092. ),
2093. child: Column(
2094. children: [
2095. SectionHeader(
2096. title: 'Book an Appointment',
2097. subtitle:
2098. 'Schedule your visit with our expert medical professionals.',
2099. ),
2100. SizedBox(height: isMobile ? 12 : 24),
2101. Card(
2102. child: Padding(
2103. padding: EdgeInsets.all(isMobile ? 16 : 24),
2104. child: Column(
2105. children: [
2106. Row(
2107. children: [
2108. Expanded(
2109. child: TabButton(
2110. label: 'New Appointment',
2111. active: tabIndex == 0,
2112. onTap: () => setState(() => tabIndex = 0),
2113. ),
2114. ),
2115. Expanded(
2116. child: TabButton(
2117. label: 'History',
2118. active: tabIndex == 1,
2119. onTap: () => setState(() => tabIndex = 1),
2120. ),
2121. ),
2122. ],
2123. ),
2124. SizedBox(height: isMobile ? 16 : 24),
2125. if (!authService.isLoggedIn && tabIndex == 0)
2126. Column(
2127. children: [
2128. Icon(
2129. Icons.lock,
2130. size: isMobile ? 32 : 40,
2131. color: Colors.indigo[700],
2132. ),
2133. SizedBox(height: 12),
2134. Text(
2135. 'Please login or sign up to book an appointment',
2136. style: GoogleFonts.poppins(
2137. fontSize: isMobile ? 16 : 18,
2138. fontWeight: FontWeight.w500,
2139. ),
2140. textAlign: TextAlign.center,
2141. ),
2142. SizedBox(height: 16),
2143. ElevatedButton(
2144. onPressed: () {
2145. showDialog(
2146. context: context,
2147. builder: (context) => AuthDialog(),
2148. );
2149. },
2150. child: Text('Login / Sign Up'),
2151. ),
2152. ],
2153. )
2154. else
2155. tabIndex == 0
2156. ? AppointmentForm(isMobile: isMobile)
2157. : AppointmentHistory(isMobile: isMobile),
2158. ],
2159. ),
2160. ),
2161. ),
2162. ],
2163. ),
2164. );
2165. },
2166. );
2167. },
2168. );
2169. }
2170. }
2171. class TabButton extends StatelessWidget {
2172. final String label;
2173. final bool active;
2174. final VoidCallback onTap;
2175. TabButton({required this.label, required this.active, required this.onTap});
2176. @override
2177. Widget build(BuildContext context) {
2178. return AnimatedContainer(
2179. duration: Duration(milliseconds: 300),
2180. decoration: BoxDecoration(
2181. color: active ? Colors.indigo[700] : Color(0xFFF8FAFC),
2182. borderRadius: BorderRadius.circular(12),
2183. boxShadow: active
2184. ? [
2185. BoxShadow(
2186. color: Colors.indigo.withOpacity(0.3),
2187. blurRadius: 4,
2188. offset: Offset(0, 2),
2189. ),
2190. ]
2191. : [],
2192. ),
2193. child: TextButton(
2194. style: TextButton.styleFrom(
2195. foregroundColor: active ? Colors.white : Color(0xFF6B7280),
2196. padding: EdgeInsets.symmetric(vertical: 12),
2197. ),
2198. onPressed: onTap,
2199. child: Text(
2200. label,
2201. style: GoogleFonts.poppins(
2202. color: active ? Colors.white : Color(0xFF6B7280),
2203. fontWeight: FontWeight.w500,
2204. ),
2205. ),
2206. ),
2207. );
2208. }
2209. }
2210. class AppointmentForm extends StatefulWidget {
2211. final bool isMobile;
2212. AppointmentForm({required this.isMobile});
2213. @override
2214. State<AppointmentForm> createState() => \_AppointmentFormState();
2215. }
2216. class \_AppointmentFormState extends State<AppointmentForm> {
2217. DateTime? selectedDate;
2218. String? selectedTime;
2219. String? department;
2220. String? doctor;
2221. String? appointmentType;
2222. final reasonController = TextEditingController();
2223. final departments = [
2224. 'Cardiology',
2225. 'Neurology',
2226. 'Orthopedics',
2227. 'Pediatrics',
2228. 'Dermatology',
2229. ];
2230. final doctorsByDepartment = {
2231. 'Cardiology': [
2232. 'Dra. Joan Dympha P. Reano',
2233. 'Dr. Rocky Danillo S. Willis',
2234. 'Dr. James Patrick Diaz',
2235. ],
2236. 'Neurology': [
2237. 'Dra. Cynthia B. Anacay',
2238. 'Dr. Mark Kryster I. Panaligan',
2239. 'Dr. Roman Santos',
2240. ],
2241. 'Orthopedics': [
2242. 'Dra. Arabella P. Quing',
2243. 'Dra. Maria Thea M. Nido',
2244. 'Dr. Edgardo J. Cuadra',
2245. ],
2246. 'Pediatrics': [
2247. 'Dr. Anner A. Marquez',
2248. 'Dr. Ching C. Casao',
2249. 'Dr. Jaypee V. Perez',
2250. ],
2251. 'Dermatology': [
2252. 'Dra. Ashley R. Medina',
2253. 'Dra. Maria Reginas B. Evia',
2254. 'Dr. Gay C. San Antonio',
2255. ],
2256. };
2257. final appointmentTypes = [
2258. 'New Patient',
2259. 'Follow-up',
2260. 'Consultation',
2261. 'Procedure',
2262. ];
2263. List<String> get doctors =>
2264. department != null && doctorsByDepartment.containsKey(department)
2265. ? doctorsByDepartment[department]!
2266. : [];
2267. List<String> get timeSlots {
2268. final slots = <String>[];
2269. for (int h = 9; h <= 17; h++) {
2270. slots.add('${\_formatHour(h)}:00 ${\_ampm(h)}');
2271. if (h < 17) slots.add('${\_formatHour(h)}:30 ${\_ampm(h)}');
2272. }
2273. return slots;
2274. }
2275. static String \_formatHour(int h) => (h % 12 == 0 ? 12 : h % 12).toString();
2276. static String \_ampm(int h) => h < 12 ? 'AM' : 'PM';
2277. void \_submit() {
2278. String? errorMessage;
2279. if (selectedDate == null) {
2280. errorMessage = 'Please select a date.';
2281. } else if (selectedTime == null) {
2282. errorMessage = 'Please select a time slot.';
2283. } else if (department == null) {
2284. errorMessage = 'Please select a department.';
2285. } else if (doctor == null) {
2286. errorMessage = 'Please select a doctor.';
2287. } else if (appointmentType == null) {
2288. errorMessage = 'Please select an appointment type.';
2289. }
2290. if (errorMessage != null) {
2291. ScaffoldMessenger.of(
2292. context,
2293. ).showSnackBar(SnackBar(content: Text(errorMessage)));
2294. return;
2295. }
2296. final appointment = {
2297. 'avatar':
2298. 'https://api.dicebear.com/7.x/avataaars/svg?seed=${doctor!.replaceAll(' ', '')}',
2299. 'doctor': doctor,
2300. 'department': department,
2301. 'type': appointmentType,
2302. 'date':
2303. DateFormat('MMM d, yyyy').format(selectedDate!) + ' at $selectedTime',
2304. 'status': 'Upcoming',
2305. 'reason': reasonController.text,
2306. };
2307. Provider.of<AppointmentService>(context, listen: false)
2308. .saveAppointment(appointment)
2309. .then((\_) {
2310. if (mounted) {
2311. ScaffoldMessenger.of(context).showSnackBar(
2312. SnackBar(content: Text('Appointment scheduled successfully!')),
2313. );
2314. Navigator.pushNamed(
2315. context,
2316. '/appointment-receipt',
2317. arguments: {
2318. 'date': selectedDate,
2319. 'time': selectedTime,
2320. 'doctor': doctor,
2321. 'department': department,
2322. 'appointmentType': appointmentType,
2323. 'reason': reasonController.text,
2324. 'username':
2325. Provider.of<AuthService>(context, listen: false).username ??
2326. 'Guest',
2327. },
2328. );
2329. setState(() {
2330. selectedDate = null;
2331. selectedTime = null;
2332. department = null;
2333. doctor = null;
2334. appointmentType = null;
2335. reasonController.clear();
2336. });
2337. }
2338. })
2339. .catchError((e) {
2340. if (mounted) {
2341. ScaffoldMessenger.of(context).showSnackBar(
2342. SnackBar(content: Text('Failed to save appointment: $e')),
2343. );
2344. }
2345. });
2346. }
2347. @override
2348. Widget build(BuildContext context) {
2349. return Builder(
2350. builder: (buildContext) => widget.isMobile
2351. ? Column(
2352. crossAxisAlignment: CrossAxisAlignment.start,
2353. children: [
2354. Text(
2355. 'Select Date & Time',
2356. style: GoogleFonts.poppins(
2357. fontSize: widget.isMobile ? 16 : 18,
2358. fontWeight: FontWeight.w500,
2359. ),
2360. ),
2361. SizedBox(height: 12),
2362. CalendarDatePicker(
2363. initialDate: DateTime.now(),
2364. firstDate: DateTime.now(),
2365. lastDate: DateTime.now().add(Duration(days: 365)),
2366. onDateChanged: (date) => setState(() {
2367. selectedDate = date;
2368. selectedTime = null;
2369. }),
2370. ),
2371. SizedBox(height: 12),
2372. Text(
2373. 'Available Time Slots',
2374. style: GoogleFonts.poppins(
2375. fontSize: widget.isMobile ? 16 : 18,
2376. fontWeight: FontWeight.w500,
2377. ),
2378. ),
2379. SizedBox(height: 8),
2380. Wrap(
2381. spacing: 8,
2382. runSpacing: 8,
2383. children: timeSlots
2384. .map(
2385. (slot) => ChoiceChip(
2386. label: Text(slot),
2387. selected: selectedTime == slot,
2388. onSelected: (\_) =>
2389. setState(() => selectedTime = slot),
2390. selectedColor: Colors.indigo[700],
2391. labelStyle: GoogleFonts.poppins(
2392. color: selectedTime == slot
2393. ? Colors.white
2394. : Colors.indigo[700],
2395. fontSize: widget.isMobile ? 14 : 16,
2396. ),
2397. backgroundColor: Colors.indigo.withOpacity(0.05),
2398. side: BorderSide(color: Colors.indigo[700]!),
2399. padding: EdgeInsets.symmetric(
2400. horizontal: widget.isMobile ? 12 : 16,
2401. vertical: widget.isMobile ? 6 : 8,
2402. ),
2403. elevation: selectedTime == slot ? 2 : 0,
2404. ),
2405. )
2406. .toList(),
2407. ),
2408. SizedBox(height: 16),
2409. Text(
2410. 'Appointment Details',
2411. style: GoogleFonts.poppins(
2412. fontSize: widget.isMobile ? 16 : 18,
2413. fontWeight: FontWeight.w500,
2414. ),
2415. ),
2416. SizedBox(height: 12),
2417. DropdownButtonFormField<String>(
2418. decoration: InputDecoration(
2419. labelText: 'Department',
2420. border: OutlineInputBorder(
2421. borderRadius: BorderRadius.circular(12),
2422. ),
2423. filled: true,
2424. fillColor: Colors.indigo.withOpacity(0.05),
2425. ),
2426. value: department,
2427. items: [
2428. DropdownMenuItem(
2429. child: Text('Select Department'),
2430. value: null,
2431. ),
2432. ...departments.map(
2433. (d) => DropdownMenuItem(child: Text(d), value: d),
2434. ),
2435. ],
2436. onChanged: (v) => setState(() {
2437. department = v;
2438. doctor = null;
2439. }),
2440. ),
2441. SizedBox(height: 12),
2442. DropdownButtonFormField<String>(
2443. decoration: InputDecoration(
2444. labelText: 'Doctor',
2445. border: OutlineInputBorder(
2446. borderRadius: BorderRadius.circular(12),
2447. ),
2448. filled: true,
2449. fillColor: Colors.indigo.withOpacity(0.05),
2450. ),
2451. value: doctor,
2452. items: [
2453. DropdownMenuItem(child: Text('Select Doctor'), value: null),
2454. ...doctors.map(
2455. (d) => DropdownMenuItem(child: Text(d), value: d),
2456. ),
2457. ],
2458. onChanged: (v) => setState(() => doctor = v),
2459. ),
2460. SizedBox(height: 12),
2461. DropdownButtonFormField<String>(
2462. decoration: InputDecoration(
2463. labelText: 'Appointment Type',
2464. border: OutlineInputBorder(
2465. borderRadius: BorderRadius.circular(12),
2466. ),
2467. filled: true,
2468. fillColor: Colors.indigo.withOpacity(0.05),
2469. ),
2470. value: appointmentType,
2471. items: [
2472. DropdownMenuItem(child: Text('Select Type'), value: null),
2473. ...appointmentTypes.map(
2474. (t) => DropdownMenuItem(child: Text(t), value: t),
2475. ),
2476. ],
2477. onChanged: (v) => setState(() => appointmentType = v),
2478. ),
2479. SizedBox(height: 12),
2480. TextFormField(
2481. controller: reasonController,
2482. decoration: InputDecoration(
2483. labelText: 'Reason for Visit',
2484. hintText:
2485. 'Please describe your symptoms or reason for visit.',
2486. border: OutlineInputBorder(
2487. borderRadius: BorderRadius.circular(12),
2488. ),
2489. filled: true,
2490. fillColor: Colors.indigo.withOpacity(0.05),
2491. ),
2492. minLines: 3,
2493. maxLines: 5,
2494. ),
2495. SizedBox(height: 16),
2496. ElevatedButton(
2497. onPressed: \_submit,
2498. child: Text('Schedule Appointment'),
2499. ),
2500. ],
2501. )
2502. : Row(
2503. crossAxisAlignment: CrossAxisAlignment.start,
2504. children: [
2505. Expanded(
2506. child: Column(
2507. crossAxisAlignment: CrossAxisAlignment.start,
2508. children: [
2509. Text(
2510. 'Select Date & Time',
2511. style: GoogleFonts.poppins(
2512. fontSize: 18,
2513. fontWeight: FontWeight.w500,
2514. ),
2515. ),
2516. SizedBox(height: 12),
2517. CalendarDatePicker(
2518. initialDate: DateTime.now(),
2519. firstDate: DateTime.now(),
2520. lastDate: DateTime.now().add(Duration(days: 365)),
2521. onDateChanged: (date) => setState(() {
2522. selectedDate = date;
2523. selectedTime = null;
2524. }),
2525. ),
2526. SizedBox(height: 12),
2527. Text(
2528. 'Available Time Slots',
2529. style: GoogleFonts.poppins(
2530. fontSize: 18,
2531. fontWeight: FontWeight.w500,
2532. ),
2533. ),
2534. SizedBox(height: 8),
2535. Wrap(
2536. spacing: 8,
2537. runSpacing: 8,
2538. children: timeSlots
2539. .map(
2540. (slot) => ChoiceChip(
2541. label: Text(slot),
2542. selected: selectedTime == slot,
2543. onSelected: (\_) =>
2544. setState(() => selectedTime = slot),
2545. selectedColor: Colors.indigo[700],
2546. labelStyle: GoogleFonts.poppins(
2547. color: selectedTime == slot
2548. ? Colors.white
2549. : Colors.indigo[700],
2550. ),
2551. backgroundColor: Colors.indigo.withOpacity(
2552. 0.05,
2553. ),
2554. side: BorderSide(color: Colors.indigo[700]!),
2555. padding: EdgeInsets.symmetric(
2556. horizontal: 16,
2557. vertical: 8,
2558. ),
2559. elevation: selectedTime == slot ? 2 : 0,
2560. ),
2561. )
2562. .toList(),
2563. ),
2564. ],
2565. ),
2566. ),
2567. SizedBox(width: 24),
2568. Expanded(
2569. child: Column(
2570. crossAxisAlignment: CrossAxisAlignment.start,
2571. children: [
2572. Text(
2573. 'Appointment Details',
2574. style: GoogleFonts.poppins(
2575. fontSize: 18,
2576. fontWeight: FontWeight.w500,
2577. ),
2578. ),
2579. SizedBox(height: 12),
2580. DropdownButtonFormField<String>(
2581. decoration: InputDecoration(
2582. labelText: 'Department',
2583. border: OutlineInputBorder(
2584. borderRadius: BorderRadius.circular(12),
2585. ),
2586. filled: true,
2587. fillColor: Colors.indigo.withOpacity(0.05),
2588. ),
2589. value: department,
2590. items: [
2591. DropdownMenuItem(
2592. child: Text('Select Department'),
2593. value: null,
2594. ),
2595. ...departments.map(
2596. (d) => DropdownMenuItem(child: Text(d), value: d),
2597. ),
2598. ],
2599. onChanged: (v) => setState(() {
2600. department = v;
2601. doctor = null;
2602. }),
2603. ),
2604. SizedBox(height: 12),
2605. DropdownButtonFormField<String>(
2606. decoration: InputDecoration(
2607. labelText: 'Doctor',
2608. border: OutlineInputBorder(
2609. borderRadius: BorderRadius.circular(12),
2610. ),
2611. filled: true,
2612. fillColor: Colors.indigo.withOpacity(0.05),
2613. ),
2614. value: doctor,
2615. items: [
2616. DropdownMenuItem(
2617. child: Text('Select Doctor'),
2618. value: null,
2619. ),
2620. ...doctors.map(
2621. (d) => DropdownMenuItem(child: Text(d), value: d),
2622. ),
2623. ],
2624. onChanged: (v) => setState(() => doctor = v),
2625. ),
2626. SizedBox(height: 12),
2627. DropdownButtonFormField<String>(
2628. decoration: InputDecoration(
2629. labelText: 'Appointment Type',
2630. border: OutlineInputBorder(
2631. borderRadius: BorderRadius.circular(12),
2632. ),
2633. filled: true,
2634. fillColor: Colors.indigo.withOpacity(0.05),
2635. ),
2636. value: appointmentType,
2637. items: [
2638. DropdownMenuItem(
2639. child: Text('Select Type'),
2640. value: null,
2641. ),
2642. ...appointmentTypes.map(
2643. (t) => DropdownMenuItem(child: Text(t), value: t),
2644. ),
2645. ],
2646. onChanged: (v) => setState(() => appointmentType = v),
2647. ),
2648. SizedBox(height: 12),
2649. TextFormField(
2650. controller: reasonController,
2651. decoration: InputDecoration(
2652. labelText: 'Reason for Visit',
2653. hintText:
2654. 'Please describe your symptoms or reason for visit.',
2655. border: OutlineInputBorder(
2656. borderRadius: BorderRadius.circular(12),
2657. ),
2658. filled: true,
2659. fillColor: Colors.indigo.withOpacity(0.05),
2660. ),
2661. minLines: 3,
2662. maxLines: 5,
2663. ),
2664. SizedBox(height: 16),
2665. ElevatedButton(
2666. onPressed: \_submit,
2667. child: Text('Schedule Appointment'),
2668. ),
2669. ],
2670. ),
2671. ),
2672. ],
2673. ),
2674. );
2675. }
2676. }
2677. class AppointmentReceiptPage extends StatelessWidget {
2678. Future<void> \_downloadReceipt(
2679. BuildContext context, {
2680. required String id,
2681. required String? username,
2682. required DateTime? date,
2683. required String? time,
2684. required String? doctor,
2685. required String? department,
2686. required String? appointmentType,
2687. required String? reason,
2688. }) async {
2689. final pdf = pw.Document();
2690. final font = await PdfGoogleFonts.poppinsRegular();
2691. pdf.addPage(
2692. pw.Page(
2693. pageFormat: PdfPageFormat.a4,
2694. build: (pw.Context context) {
2695. return pw.Padding(
2696. padding: const pw.EdgeInsets.all(24),
2697. child: pw.Column(
2698. crossAxisAlignment: pw.CrossAxisAlignment.start,
2699. children: [
2700. pw.Row(
2701. mainAxisAlignment: pw.MainAxisAlignment.center,
2702. children: [
2703. pw.Text(
2704. 'Global Care Medical Center',
2705. style: pw.TextStyle(
2706. font: font,
2707. fontSize: 24,
2708. fontWeight: pw.FontWeight.bold,
2709. color: PdfColors.indigo900,
2710. ),
2711. ),
2712. ],
2713. ),
2714. pw.SizedBox(height: 16),
2715. pw.Text(
2716. 'Appointment Receipt',
2717. style: pw.TextStyle(
2718. font: font,
2719. fontSize: 20,
2720. fontWeight: pw.FontWeight.bold,
2721. color: PdfColors.indigo700,
2722. ),
2723. ),
2724. pw.SizedBox(height: 8),
2725. pw.Text(
2726. 'Your appointment has been successfully scheduled.',
2727. style: pw.TextStyle(
2728. font: font,
2729. fontSize: 14,
2730. color: PdfColors.grey600,
2731. ),
2732. ),
2733. pw.SizedBox(height: 24),
2734. pw.Container(
2735. padding: const pw.EdgeInsets.all(12),
2736. decoration: pw.BoxDecoration(
2737. border: pw.Border.all(color: PdfColors.indigo200),
2738. borderRadius: pw.BorderRadius.circular(12),
2739. ),
2740. child: pw.Column(
2741. crossAxisAlignment: pw.CrossAxisAlignment.start,
2742. children: [
2743. pw.Text(
2744. 'Appointment Details',
2745. style: pw.TextStyle(
2746. font: font,
2747. fontSize: 16,
2748. fontWeight: pw.FontWeight.bold,
2749. ),
2750. ),
2751. pw.SizedBox(height: 8),
2752. \_pdfSummaryRow('Appointment ID:', '#$id', font),
2753. \_pdfSummaryRow('Patient:', username ?? 'Guest', font),
2754. \_pdfSummaryRow(
2755. 'Date:',
2756. date != null
2757. ? DateFormat('MMMM d, yyyy').format(date)
2758. : 'N/A',
2759. font,
2760. ),
2761. \_pdfSummaryRow('Time:', time ?? 'N/A', font),
2762. \_pdfSummaryRow('Doctor:', doctor ?? 'N/A', font),
2763. \_pdfSummaryRow('Department:', department ?? 'N/A', font),
2764. \_pdfSummaryRow('Type:', appointmentType ?? 'N/A', font),
2765. \_pdfSummaryRow(
2766. 'Reason:',
2767. reason?.isEmpty ?? true ? 'Not specified' : reason!,
2768. font,
2769. ),
2770. ],
2771. ),
2772. ),
2773. pw.SizedBox(height: 16),
2774. pw.Container(
2775. padding: const pw.EdgeInsets.only(left: 12),
2776. decoration: const pw.BoxDecoration(
2777. border: pw.Border(
2778. left: pw.BorderSide(color: PdfColors.indigo700, width: 3),
2779. ),
2780. ),
2781. child: pw.Column(
2782. crossAxisAlignment: pw.CrossAxisAlignment.start,
2783. children: [
2784. pw.Text(
2785. 'Confirmation',
2786. style: pw.TextStyle(
2787. font: font,
2788. fontSize: 16,
2789. fontWeight: pw.FontWeight.bold,
2790. ),
2791. ),
2792. pw.SizedBox(height: 8),
2793. pw.Text(
2794. 'A confirmation email will be sent to your registered email address.',
2795. style: pw.TextStyle(font: font, fontSize: 12),
2796. ),
2797. pw.SizedBox(height: 8),
2798. pw.Text(
2799. 'Please arrive 15 minutes early and bring a valid ID.',
2800. style: pw.TextStyle(font: font, fontSize: 12),
2801. ),
2802. pw.SizedBox(height: 8),
2803. pw.Text(
2804. 'Location: J. Yulo Avenue, Brgy. Canlubang',
2805. style: pw.TextStyle(font: font, fontSize: 12),
2806. ),
2807. ],
2808. ),
2809. ),
2810. ],
2811. ),
2812. );
2813. },
2814. ),
2815. );
2816. await Printing.sharePdf(
2817. bytes: await pdf.save(),
2818. filename: 'appointment\_receipt\_$id.pdf',
2819. );
2820. }
2821. pw.Widget \_pdfSummaryRow(String label, String value, pw.Font font) {
2822. return pw.Padding(
2823. padding: const pw.EdgeInsets.symmetric(vertical: 4),
2824. child: pw.Row(
2825. crossAxisAlignment: pw.CrossAxisAlignment.start,
2826. children: [
2827. pw.SizedBox(
2828. width: 140,
2829. child: pw.Text(
2830. label,
2831. style: pw.TextStyle(
2832. font: font,
2833. fontSize: 12,
2834. fontWeight: pw.FontWeight.bold,
2835. color: PdfColors.grey600,
2836. ),
2837. ),
2838. ),
2839. pw.Expanded(
2840. child: pw.Text(
2841. value,
2842. style: pw.TextStyle(font: font, fontSize: 12),
2843. ),
2844. ),
2845. ],
2846. ),
2847. );
2848. }
2849. @override
2850. Widget build(BuildContext context) {
2851. final Map<String, dynamic>? args =
2852. ModalRoute.of(context)?.settings.arguments as Map<String, dynamic>?;
2853. final DateTime? date = args?['date'] as DateTime?;
2854. final String? time = args?['time'] as String?;
2855. final String? doctor = args?['doctor'] as String?;
2856. final String? department = args?['department'] as String?;
2857. final String? appointmentType = args?['appointmentType'] as String?;
2858. final String? reason = args?['reason'] as String?;
2859. final String? username = args?['username'] as String?;
2860. final id = date != null
2861. ? (100000 + (date.millisecondsSinceEpoch % 900000)).toString()
2862. : 'N/A';
2863. return Scaffold(
2864. appBar: AppBar(
2865. leading: IconButton(
2866. icon: Icon(Icons.arrow\_back, color: Colors.indigo[700]),
2867. onPressed: () {
2868. Navigator.pop(context);
2869. final tabController = DefaultTabController.of(context);
2870. if (tabController != null) {
2871. tabController.animateTo(1);
2872. }
2873. },
2874. ),
2875. title: Text(
2876. 'Appointment Receipt',
2877. style: GoogleFonts.poppins(
2878. color: Colors.indigo[900],
2879. fontSize: MediaQuery.of(context).size.width < 600 ? 20 : 24,
2880. fontWeight: FontWeight.w600,
2881. ),
2882. ),
2883. backgroundColor: Colors.white,
2884. elevation: 0,
2885. shadowColor: Colors.black12,
2886. ),
2887. body: SingleChildScrollView(
2888. child: LayoutBuilder(
2889. builder: (context, constraints) {
2890. final isMobile = constraints.maxWidth < 600;
2891. return Container(
2892. color: Color(0xFFF8FAFC),
2893. padding: EdgeInsets.symmetric(
2894. vertical: isMobile ? 24 : 40,
2895. horizontal: constraints.maxWidth \* 0.05,
2896. ),
2897. child: Column(
2898. children: [
2899. SectionHeader(
2900. title: 'Appointment Receipt',
2901. subtitle:
2902. 'Your appointment has been successfully scheduled.',
2903. ),
2904. SizedBox(height: isMobile ? 12 : 24),
2905. Card(
2906. child: Padding(
2907. padding: EdgeInsets.all(isMobile ? 16 : 24),
2908. child: Column(
2909. crossAxisAlignment: CrossAxisAlignment.start,
2910. children: [
2911. Row(
2912. children: [
2913. Icon(
2914. Icons.check\_circle,
2915. color: Colors.green,
2916. size: isMobile ? 24 : 28,
2917. ),
2918. SizedBox(width: 8),
2919. Text(
2920. 'Appointment Confirmed',
2921. style: GoogleFonts.poppins(
2922. fontSize: isMobile ? 16 : 18,
2923. fontWeight: FontWeight.w500,
2924. ),
2925. ),
2926. ],
2927. ),
2928. SizedBox(height: 12),
2929. Container(
2930. decoration: BoxDecoration(
2931. color: Colors.indigo.withOpacity(0.05),
2932. borderRadius: BorderRadius.circular(12),
2933. border: Border.all(
2934. color: Colors.indigo.withOpacity(0.2),
2935. ),
2936. ),
2937. padding: EdgeInsets.all(12),
2938. width: double.infinity,
2939. child: Column(
2940. crossAxisAlignment: CrossAxisAlignment.start,
2941. children: [
2942. Text(
2943. 'Appointment Details',
2944. style: GoogleFonts.poppins(
2945. fontSize: isMobile ? 14 : 16,
2946. fontWeight: FontWeight.w500,
2947. ),
2948. ),
2949. SizedBox(height: 8),
2950. \_summaryRow(
2951. 'Appointment ID:',
2952. '#$id',
2953. isMobile,
2954. ),
2955. \_summaryRow(
2956. 'Patient:',
2957. username ?? 'Guest',
2958. isMobile,
2959. ),
2960. \_summaryRow(
2961. 'Date:',
2962. date != null
2963. ? DateFormat('MMMM d, yyyy').format(date)
2964. : 'N/A',
2965. isMobile,
2966. ),
2967. \_summaryRow('Time:', time ?? 'N/A', isMobile),
2968. \_summaryRow(
2969. 'Doctor:',
2970. doctor ?? 'N/A',
2971. isMobile,
2972. ),
2973. \_summaryRow(
2974. 'Department:',
2975. department ?? 'N/A',
2976. isMobile,
2977. ),
2978. \_summaryRow(
2979. 'Type:',
2980. appointmentType ?? 'N/A',
2981. isMobile,
2982. ),
2983. \_summaryRow(
2984. 'Reason:',
2985. reason?.isEmpty ?? true
2986. ? 'Not specified'
2987. : reason!,
2988. isMobile,
2989. ),
2990. ],
2991. ),
2992. ),
2993. SizedBox(height: 12),
2994. Container(
2995. decoration: BoxDecoration(
2996. border: Border(
2997. left: BorderSide(
2998. color: Colors.indigo[700]!,
2999. width: 3,
3000. ),
3001. ),
3002. ),
3003. padding: EdgeInsets.only(left: 12),
3004. child: Column(
3005. crossAxisAlignment: CrossAxisAlignment.start,
3006. children: [
3007. Text(
3008. 'Confirmation',
3009. style: GoogleFonts.poppins(
3010. fontSize: isMobile ? 14 : 16,
3011. fontWeight: FontWeight.w500,
3012. ),
3013. ),
3014. SizedBox(height: 8),
3015. Text(
3016. 'A confirmation email will be sent to your registered email address.',
3017. style: GoogleFonts.poppins(
3018. fontSize: isMobile ? 12 : 14,
3019. ),
3020. ),
3021. SizedBox(height: 8),
3022. Text(
3023. 'Please arrive 15 minutes early and bring a valid ID.',
3024. style: GoogleFonts.poppins(
3025. fontSize: isMobile ? 12 : 14,
3026. ),
3027. ),
3028. SizedBox(height: 8),
3029. Text(
3030. 'Location: J. Yulo Avenue, Brgy. Canlubang',
3031. style: GoogleFonts.poppins(
3032. fontSize: isMobile ? 12 : 14,
3033. ),
3034. ),
3035. ],
3036. ),
3037. ),
3038. SizedBox(height: 16),
3039. Row(
3040. mainAxisAlignment: MainAxisAlignment.end,
3041. children: [
3042. OutlinedButton(
3043. onPressed: () {
3044. Navigator.pop(context);
3045. final tabController = DefaultTabController.of(
3046. context,
3047. );
3048. if (tabController != null) {
3049. tabController.animateTo(1);
3050. }
3051. },
3052. child: Text('Back to Appointments'),
3053. ),
3054. SizedBox(width: 8),
3055. ElevatedButton.icon(
3056. onPressed: () => \_downloadReceipt(
3057. context,
3058. id: id,
3059. username: username,
3060. date: date,
3061. time: time,
3062. doctor: doctor,
3063. department: department,
3064. appointmentType: appointmentType,
3065. reason: reason,
3066. ),
3067. icon: Icon(
3068. Icons.download,
3069. size: isMobile ? 16 : 18,
3070. ),
3071. label: Text('Download Receipt'),
3072. style: ElevatedButton.styleFrom(
3073. backgroundColor: Colors.indigo[700],
3074. foregroundColor: Colors.white,
3075. padding: EdgeInsets.symmetric(
3076. horizontal: isMobile ? 12 : 16,
3077. vertical: isMobile ? 8 : 10,
3078. ),
3079. ),
3080. ),
3081. ],
3082. ),
3083. ],
3084. ),
3085. ),
3086. ),
3087. ],
3088. ),
3089. );
3090. },
3091. ),
3092. ),
3093. );
3094. }
3095. Widget \_summaryRow(String label, String value, bool isMobile) {
3096. return Padding(
3097. padding: EdgeInsets.symmetric(vertical: 4),
3098. child: Row(
3099. crossAxisAlignment: CrossAxisAlignment.start,
3100. children: [
3101. SizedBox(
3102. width: isMobile ? 120 : 140,
3103. child: Text(
3104. label,
3105. style: GoogleFonts.poppins(
3106. fontSize: isMobile ? 12 : 14,
3107. fontWeight: FontWeight.w500,
3108. color: Colors.grey[600],
3109. ),
3110. ),
3111. ),
3112. Expanded(
3113. child: Text(
3114. value,
3115. style: GoogleFonts.poppins(
3116. fontSize: isMobile ? 12 : 14,
3117. fontWeight: FontWeight.w400,
3118. ),
3119. ),
3120. ),
3121. ],
3122. ),
3123. );
3124. }
3125. }
3126. class AppointmentHistory extends StatelessWidget {
3127. final bool isMobile;
3128. AppointmentHistory({required this.isMobile}); // Fixed constructor
3129. @override
3130. Widget build(BuildContext context) {
3131. return FutureBuilder<List<Map<String, dynamic>>>(
3132. future: Provider.of<AppointmentService>(context).loadAppointments(),
3133. builder: (context, snapshot) {
3134. if (snapshot.connectionState == ConnectionState.waiting) {
3135. return Center(child: CircularProgressIndicator());
3136. }
3137. if (snapshot.hasError) {
3138. return Center(child: Text('Error loading appointments'));
3139. }
3140. final appointments = snapshot.data ?? [];
3141. if (appointments.isEmpty) {
3142. return Center(
3143. child: Text(
3144. 'No appointments found.',
3145. style: GoogleFonts.poppins(fontSize: isMobile ? 16 : 18),
3146. ),
3147. );
3148. }
3149. return ListView.separated(
3150. shrinkWrap: true,
3151. physics: NeverScrollableScrollPhysics(),
3152. itemCount: appointments.length,
3153. separatorBuilder: (context, index) => SizedBox(height: 12),
3154. itemBuilder: (context, index) {
3155. final appointment = appointments[index];
3156. return Card(
3157. child: Padding(
3158. padding: EdgeInsets.all(isMobile ? 12 : 16),
3159. child: Row(
3160. crossAxisAlignment: CrossAxisAlignment.start,
3161. children: [
3162. CircleAvatar(
3163. backgroundImage: NetworkImage(appointment['avatar']),
3164. radius: isMobile ? 24 : 28,
3165. ),
3166. SizedBox(width: 12),
3167. Expanded(
3168. child: Column(
3169. crossAxisAlignment: CrossAxisAlignment.start,
3170. children: [
3171. Text(
3172. appointment['doctor'] ?? 'N/A',
3173. style: GoogleFonts.poppins(
3174. fontSize: isMobile ? 16 : 18,
3175. fontWeight: FontWeight.w500,
3176. ),
3177. ),
3178. SizedBox(height: 4),
3179. Text(
3180. appointment['department'] ?? 'N/A',
3181. style: GoogleFonts.poppins(
3182. fontSize: isMobile ? 14 : 16,
3183. color: Colors.grey[600],
3184. ),
3185. ),
3186. SizedBox(height: 4),
3187. Text(
3188. appointment['date'] ?? 'N/A',
3189. style: GoogleFonts.poppins(
3190. fontSize: isMobile ? 14 : 16,
3191. color: Colors.grey[600],
3192. ),
3193. ),
3194. SizedBox(height: 4),
3195. Text(
3196. appointment['type'] ?? 'N/A',
3197. style: GoogleFonts.poppins(
3198. fontSize: isMobile ? 14 : 16,
3199. color: Colors.grey[600],
3200. ),
3201. ),
3202. SizedBox(height: 4),
3203. Text(
3204. 'Status: ${appointment['status'] ?? 'N/A'}',
3205. style: GoogleFonts.poppins(
3206. fontSize: isMobile ? 14 : 16,
3207. color: appointment['status'] == 'Upcoming'
3208. ? Colors.green
3209. : Colors.grey[600],
3210. ),
3211. ),
3212. if (appointment['reason']?.isNotEmpty ?? false)
3213. Column(
3214. crossAxisAlignment: CrossAxisAlignment.start,
3215. children: [
3216. SizedBox(height: 8),
3217. Text(
3218. 'Reason: ${appointment['reason']}',
3219. style: GoogleFonts.poppins(
3220. fontSize: isMobile ? 14 : 16,
3221. color: Colors.grey[600],
3222. ),
3223. ),
3224. ],
3225. ),
3226. ],
3227. ),
3228. ),
3229. ],
3230. ),
3231. ),
3232. );
3233. },
3234. );
3235. },
3236. );
3237. }
3238. }
3239. class ContactSection extends StatelessWidget {
3240. @override
3241. Widget build(BuildContext context) {
3242. return LayoutBuilder(
3243. builder: (context, constraints) {
3244. final isMobile = constraints.maxWidth < 600;
3245. return Container(
3246. color: Color(0xFFF8FAFC),
3247. padding: EdgeInsets.symmetric(
3248. vertical: isMobile ? 24 : 40,
3249. horizontal: constraints.maxWidth \* 0.05,
3250. ),
3251. child: Column(
3252. children: [
3253. SectionHeader(
3254. title: 'Contact Us',
3255. subtitle: 'Get in touch with our team for any inquiries.',
3256. ),
3257. SizedBox(height: isMobile ? 12 : 24),
3258. Card(
3259. child: Padding(
3260. padding: EdgeInsets.all(isMobile ? 16 : 24),
3261. child: isMobile
3262. ? Column(
3263. crossAxisAlignment: CrossAxisAlignment.start,
3264. children: [
3265. ContactItem(
3266. icon: Icons.location\_on,
3267. title: 'Address',
3268. content: 'J. Yulo Avenue, Brgy. Canlubang',
3269. isMobile: isMobile,
3270. ),
3271. SizedBox(height: 16),
3272. ContactItem(
3273. icon: Icons.phone,
3274. title: 'Phone',
3275. content: '(049) 520-5626',
3276. isMobile: isMobile,
3277. onTap: () async {
3278. final url = Uri.parse('tel:(049) 520-5626');
3279. if (await canLaunchUrl(url)) {
3280. await launchUrl(url);
3281. } else {
3282. ScaffoldMessenger.of(context).showSnackBar(
3283. SnackBar(
3284. content: Text('Cannot launch phone'),
3285. ),
3286. );
3287. }
3288. },
3289. ),
3290. SizedBox(height: 16),
3291. ContactItem(
3292. icon: Icons.email,
3293. title: 'Email',
3294. content: 'gcmccanlubang@gmail.com',
3295. isMobile: isMobile,
3296. onTap: () async {
3297. final url = Uri.parse(
3298. 'mailto:gcmccanlubang@gmail.com',
3299. );
3300. if (await canLaunchUrl(url)) {
3301. await launchUrl(url);
3302. } else {
3303. ScaffoldMessenger.of(context).showSnackBar(
3304. SnackBar(
3305. content: Text('Cannot launch email'),
3306. ),
3307. );
3308. }
3309. },
3310. ),
3311. ],
3312. )
3313. : Row(
3314. crossAxisAlignment: CrossAxisAlignment.start,
3315. children: [
3316. Expanded(
3317. child: ContactItem(
3318. icon: Icons.location\_on,
3319. title: 'Address',
3320. content: 'J. Yulo Avenue, Brgy. Canlubang',
3321. isMobile: isMobile,
3322. ),
3323. ),
3324. SizedBox(width: 24),
3325. Expanded(
3326. child: ContactItem(
3327. icon: Icons.phone,
3328. title: 'Phone',
3329. content: '(049) 520-5626',
3330. isMobile: isMobile,
3331. onTap: () async {
3332. final url = Uri.parse('tel:(049) 520-5626');
3333. if (await canLaunchUrl(url)) {
3334. await launchUrl(url);
3335. } else {
3336. ScaffoldMessenger.of(context).showSnackBar(
3337. SnackBar(
3338. content: Text('Cannot launch phone'),
3339. ),
3340. );
3341. }
3342. },
3343. ),
3344. ),
3345. SizedBox(width: 24),
3346. Expanded(
3347. child: ContactItem(
3348. icon: Icons.email,
3349. title: 'Email',
3350. content: 'gcmccanlubang@gmail.com',
3351. isMobile: isMobile,
3352. onTap: () async {
3353. final url = Uri.parse(
3354. 'mailto:gcmccanlubang@gmail.com',
3355. );
3356. if (await canLaunchUrl(url)) {
3357. await launchUrl(url);
3358. } else {
3359. ScaffoldMessenger.of(context).showSnackBar(
3360. SnackBar(
3361. content: Text('Cannot launch email'),
3362. ),
3363. );
3364. }
3365. },
3366. ),
3367. ),
3368. ],
3369. ),
3370. ),
3371. ),
3372. ],
3373. ),
3374. );
3375. },
3376. );
3377. }
3378. }
3379. class ContactItem extends StatelessWidget {
3380. final IconData icon;
3381. final String title;
3382. final String content;
3383. final bool isMobile;
3384. final VoidCallback? onTap;
3385. ContactItem({
3386. required this.icon,
3387. required this.title,
3388. required this.content,
3389. required this.isMobile,
3390. this.onTap,
3391. });
3392. @override
3393. Widget build(BuildContext context) {
3394. return InkWell(
3395. onTap: onTap,
3396. child: Row(
3397. crossAxisAlignment: CrossAxisAlignment.start,
3398. children: [
3399. CircleAvatar(
3400. backgroundColor: Colors.indigo.withOpacity(0.1),
3401. child: Icon(icon, color: Colors.indigo[700]),
3402. radius: isMobile ? 20 : 24,
3403. ),
3404. SizedBox(width: 12),
3405. Column(
3406. crossAxisAlignment: CrossAxisAlignment.start,
3407. children: [
3408. Text(
3409. title,
3410. style: GoogleFonts.poppins(
3411. fontSize: isMobile ? 14 : 16,
3412. fontWeight: FontWeight.w500,
3413. ),
3414. ),
3415. SizedBox(height: 4),
3416. Text(
3417. content,
3418. style: GoogleFonts.poppins(
3419. fontSize: isMobile ? 12 : 14,
3420. color: Colors.grey[600],
3421. ),
3422. ),
3423. ],
3424. ),
3425. ],
3426. ),
3427. );
3428. }
3429. }
3430. class FooterSection extends StatelessWidget {
3431. @override
3432. Widget build(BuildContext context) {
3433. return LayoutBuilder(
3434. builder: (context, constraints) {
3435. final isMobile = constraints.maxWidth < 600;
3436. return Container(
3437. color: Colors.indigo[900],
3438. padding: EdgeInsets.symmetric(
3439. vertical: isMobile ? 24 : 40,
3440. horizontal: constraints.maxWidth \* 0.05,
3441. ),
3442. child: isMobile
3443. ? Column(
3444. crossAxisAlignment: CrossAxisAlignment.start,
3445. children: [
3446. Row(
3447. children: [
3448. Icon(
3449. Icons.favorite,
3450. color: Colors.white,
3451. size: isMobile ? 24 : 28,
3452. ),
3453. SizedBox(width: 8),
3454. Text(
3455. 'Global Care',
3456. style: GoogleFonts.poppins(
3457. color: Colors.white,
3458. fontSize: isMobile ? 20 : 24,
3459. fontWeight: FontWeight.w700,
3460. ),
3461. ),
3462. ],
3463. ),
3464. SizedBox(height: 16),
3465. Text(
3466. 'Providing compassionate care and cutting-edge medical services to our community.',
3467. style: GoogleFonts.poppins(
3468. color: Colors.white.withOpacity(0.9),
3469. fontSize: isMobile ? 12 : 14,
3470. ),
3471. ),
3472. SizedBox(height: 24),
3473. FooterLink(
3474. text: 'Home',
3475. onTap: () {
3476. final tabController = DefaultTabController.of(context);
3477. if (tabController != null) {
3478. tabController.animateTo(0);
3479. (context as Element).markNeedsBuild();
3480. }
3481. },
3482. ),
3483. FooterLink(
3484. text: 'Appointments',
3485. onTap: () {
3486. final tabController = DefaultTabController.of(context);
3487. if (tabController != null) {
3488. tabController.animateTo(1);
3489. (context as Element).markNeedsBuild();
3490. }
3491. },
3492. ),
3493. FooterLink(
3494. text: 'Feedback',
3495. onTap: () {
3496. final tabController = DefaultTabController.of(context);
3497. if (tabController != null) {
3498. tabController.animateTo(2);
3499. (context as Element).markNeedsBuild();
3500. }
3501. },
3502. ),
3503. FooterLink(
3504. text: 'Contact Us',
3505. onTap: () {
3506. // TODO: Implement scroll to contact section
3507. ScaffoldMessenger.of(context).showSnackBar(
3508. SnackBar(content: Text('Contact Us clicked!')),
3509. );
3510. },
3511. ),
3512. SizedBox(height: 24),
3513. Text(
3514. '© 2025 Global Care Medical Center. All rights reserved.',
3515. style: GoogleFonts.poppins(
3516. color: Colors.white.withOpacity(0.7),
3517. fontSize: isMobile ? 12 : 14,
3518. ),
3519. ),
3520. ],
3521. )
3522. : Column(
3523. children: [
3524. Row(
3525. crossAxisAlignment: CrossAxisAlignment.start,
3526. children: [
3527. Expanded(
3528. flex: 2,
3529. child: Column(
3530. crossAxisAlignment: CrossAxisAlignment.start,
3531. children: [
3532. Row(
3533. children: [
3534. Icon(
3535. Icons.favorite,
3536. color: Colors.white,
3537. size: 28,
3538. ),
3539. SizedBox(width: 8),
3540. Text(
3541. 'Global Care',
3542. style: GoogleFonts.poppins(
3543. color: Colors.white,
3544. fontSize: 24,
3545. fontWeight: FontWeight.w700,
3546. ),
3547. ),
3548. ],
3549. ),
3550. SizedBox(height: 16),
3551. Text(
3552. 'Providing compassionate care and cutting-edge medical services to our community.',
3553. style: GoogleFonts.poppins(
3554. color: Colors.white.withOpacity(0.9),
3555. fontSize: 14,
3556. ),
3557. ),
3558. ],
3559. ),
3560. ),
3561. SizedBox(width: 24),
3562. Expanded(
3563. child: Column(
3564. crossAxisAlignment: CrossAxisAlignment.start,
3565. children: [
3566. Text(
3567. 'Quick Links',
3568. style: GoogleFonts.poppins(
3569. color: Colors.white,
3570. fontSize: 16,
3571. fontWeight: FontWeight.w500,
3572. ),
3573. ),
3574. SizedBox(height: 12),
3575. FooterLink(
3576. text: 'Home',
3577. onTap: () {
3578. final tabController = DefaultTabController.of(
3579. context,
3580. );
3581. if (tabController != null) {
3582. tabController.animateTo(0);
3583. (context as Element).markNeedsBuild();
3584. }
3585. },
3586. ),
3587. FooterLink(
3588. text: 'Appointments',
3589. onTap: () {
3590. final tabController = DefaultTabController.of(
3591. context,
3592. );
3593. if (tabController != null) {
3594. tabController.animateTo(1);
3595. (context as Element).markNeedsBuild();
3596. }
3597. },
3598. ),
3599. FooterLink(
3600. text: 'Feedback',
3601. onTap: () {
3602. final tabController = DefaultTabController.of(
3603. context,
3604. );
3605. if (tabController != null) {
3606. tabController.animateTo(2);
3607. (context as Element).markNeedsBuild();
3608. }
3609. },
3610. ),
3611. FooterLink(
3612. text: 'Contact Us',
3613. onTap: () {
3614. // TODO: Implement scroll to contact section
3615. ScaffoldMessenger.of(context).showSnackBar(
3616. SnackBar(
3617. content: Text('Contact Us clicked!'),
3618. ),
3619. );
3620. },
3621. ),
3622. ],
3623. ),
3624. ),
3625. SizedBox(width: 24),
3626. Expanded(
3627. child: Column(
3628. crossAxisAlignment: CrossAxisAlignment.start,
3629. children: [
3630. Text(
3631. 'Contact',
3632. style: GoogleFonts.poppins(
3633. color: Colors.white,
3634. fontSize: 16,
3635. fontWeight: FontWeight.w500,
3636. ),
3637. ),
3638. SizedBox(height: 12),
3639. FooterLink(
3640. text: 'J. Yulo Avenue, Brgy. Canlubang',
3641. icon: Icons.location\_on,
3642. ),
3643. FooterLink(
3644. text: '(049) 520-5626',
3645. icon: Icons.phone,
3646. onTap: () async {
3647. final url = Uri.parse('tel:(049) 520-5626');
3648. if (await canLaunchUrl(url)) {
3649. await launchUrl(url);
3650. } else {
3651. ScaffoldMessenger.of(context).showSnackBar(
3652. SnackBar(
3653. content: Text('Cannot launch phone'),
3654. ),
3655. );
3656. }
3657. },
3658. ),
3659. FooterLink(
3660. text: 'gcmccanlubang@gmail.com',
3661. icon: Icons.email,
3662. onTap: () async {
3663. final url = Uri.parse(
3664. 'mailto:gcmccanlubang@gmail.com',
3665. );
3666. if (await canLaunchUrl(url)) {
3667. await launchUrl(url);
3668. } else {
3669. ScaffoldMessenger.of(context).showSnackBar(
3670. SnackBar(
3671. content: Text('Cannot launch email'),
3672. ),
3673. );
3674. }
3675. },
3676. ),
3677. ],
3678. ),
3679. ),
3680. ],
3681. ),
3682. SizedBox(height: 16),
3683. Divider(color: Colors.white.withOpacity(0.3)),
3684. SizedBox(height: 16),
3685. Text(
3686. '© 2025 Global Care Medical Center. All rights reserved.',
3687. style: GoogleFonts.poppins(
3688. color: Colors.white.withOpacity(0.7),
3689. fontSize: 14,
3690. ),
3691. ),
3692. ],
3693. ),
3694. );
3695. },
3696. );
3697. }
3698. }
3699. class FooterLink extends StatelessWidget {
3700. final String text;
3701. final IconData? icon;
3702. final VoidCallback? onTap;
3703. FooterLink({required this.text, this.icon, this.onTap});
3704. @override
3705. Widget build(BuildContext context) {
3706. return InkWell(
3707. onTap: onTap,
3708. child: Padding(
3709. padding: EdgeInsets.symmetric(vertical: 6),
3710. child: Row(
3711. children: [
3712. if (icon != null) ...[
3713. Icon(icon, color: Colors.white.withOpacity(0.9), size: 16),
3714. SizedBox(width: 8),
3715. ],
3716. Text(
3717. text,
3718. style: GoogleFonts.poppins(
3719. color: Colors.white.withOpacity(0.9),
3720. fontSize: 14,
3721. decoration: onTap != null
3722. ? TextDecoration.underline
3723. : TextDecoration.none,
3724. ),
3725. ),
3726. ],
3727. ),
3728. ),
3729. );
3730. }
3731. }
3732. class SectionHeader extends StatelessWidget {
3733. final String title;
3734. final String? subtitle;
3735. SectionHeader({required this.title, this.subtitle});
3736. @override
3737. Widget build(BuildContext context) {
3738. return LayoutBuilder(
3739. builder: (context, constraints) {
3740. final isMobile = constraints.maxWidth < 600;
3741. return Column(
3742. children: [
3743. AnimatedOpacity(
3744. opacity: 1.0,
3745. duration: Duration(milliseconds: 800),
3746. child: Text(
3747. title,
3748. style: GoogleFonts.poppins(
3749. fontSize: isMobile ? 24 : 28,
3750. fontWeight: FontWeight.w600,
3751. color: Color(0xFF1F2937),
3752. ),
3753. ),
3754. ),
3755. if (subtitle != null) ...[
3756. SizedBox(height: isMobile ? 8 : 12),
3757. AnimatedOpacity(
3758. opacity: 1.0,
3759. duration: Duration(milliseconds: 1000),
3760. child: Text(
3761. subtitle!,
3762. style: GoogleFonts.poppins(
3763. fontSize: isMobile ? 14 : 16,
3764. fontWeight: FontWeight.w400,
3765. color: Color(0xFF6B7280),
3766. ),
3767. textAlign: TextAlign.center,
3768. ),
3769. ),
3770. ],
3771. ],
3772. );
3773. },
3774. );
3775. }
3776. }
3777. class DashboardSection extends StatelessWidget {
3778. @override
3779. Widget build(BuildContext context) {
3780. return Consumer<AuthService>(
3781. builder: (context, authService, child) {
3782. return LayoutBuilder(
3783. builder: (context, constraints) {
3784. final double screenWidth = constraints.maxWidth;
3785. final bool isMobile = screenWidth < 600;
3786. final bool isTablet = screenWidth >= 600 && screenWidth < 900;
3787. final double cardPadding = isMobile
3788. ? 12.0
3789. : isTablet
3790. ? 16.0
3791. : 20.0;
3792. final double sectionSpacing = isMobile
3793. ? 12.0
3794. : isTablet
3795. ? 16.0
3796. : 20.0;
3797. final double chartHeight = isMobile
3798. ? 180.0
3799. : isTablet
3800. ? 220.0
3801. : 260.0;
3802. final double fontSizeTitle = isMobile
3803. ? 16.0
3804. : isTablet
3805. ? 18.0
3806. : 20.0;
3807. final double fontSizeSubtitle = isMobile
3808. ? 12.0
3809. : isTablet
3810. ? 14.0
3811. : 16.0;
3812. return Container(
3813. color: const Color(0xFFF8FAFC),
3814. padding: EdgeInsets.symmetric(
3815. vertical: isMobile
3816. ? 16.0
3817. : isTablet
3818. ? 24.0
3819. : 32.0,
3820. horizontal: screenWidth \* 0.05,
3821. ),
3822. child: SingleChildScrollView(
3823. child: Column(
3824. crossAxisAlignment: CrossAxisAlignment.center,
3825. children: [
3826. // Welcome Banner
3827. Center(
3828. child: Container(
3829. constraints: BoxConstraints(maxWidth: 600),
3830. child: \_buildWelcomeBanner(
3831. context,
3832. authService,
3833. isMobile,
3834. fontSizeTitle,
3835. fontSizeSubtitle,
3836. ),
3837. ),
3838. ),
3839. SizedBox(height: sectionSpacing),
3840. // If not logged in
3841. if (!authService.isLoggedIn)
3842. Center(
3843. child: Container(
3844. constraints: BoxConstraints(maxWidth: 600),
3845. child: \_buildLoginPrompt(
3846. context,
3847. isMobile,
3848. cardPadding,
3849. fontSizeTitle,
3850. ),
3851. ),
3852. )
3853. else
3854. Column(
3855. crossAxisAlignment: CrossAxisAlignment.center,
3856. children: [
3857. // Overview Section
3858. Center(
3859. child: SectionHeader(
3860. title: 'Dashboard Overview',
3861. subtitle:
3862. 'Your healthcare insights and management hub.',
3863. ),
3864. ),
3865. SizedBox(height: sectionSpacing),
3866. \_buildOverviewCharts(
3867. context,
3868. isMobile,
3869. isTablet,
3870. cardPadding,
3871. chartHeight,
3872. fontSizeSubtitle,
3873. screenWidth,
3874. ),
3875. SizedBox(height: sectionSpacing),
3876. // User Profile
3877. \_buildUserProfile(
3878. context,
3879. authService,
3880. isMobile,
3881. cardPadding,
3882. fontSizeTitle,
3883. fontSizeSubtitle,
3884. ),
3885. SizedBox(height: sectionSpacing),
3886. // Appointments
3887. Text(
3888. 'Your Appointments',
3889. style: GoogleFonts.poppins(
3890. fontSize: fontSizeTitle,
3891. fontWeight: FontWeight.w600,
3892. color: const Color(0xFF1F2937),
3893. ),
3894. ),
3895. SizedBox(height: sectionSpacing / 2),
3896. AppointmentHistory(isMobile: isMobile),
3897. SizedBox(height: sectionSpacing),
3898. // Feedback
3899. Text(
3900. 'Your Feedback',
3901. style: GoogleFonts.poppins(
3902. fontSize: fontSizeTitle,
3903. fontWeight: FontWeight.w600,
3904. color: const Color(0xFF1F2937),
3905. ),
3906. ),
3907. SizedBox(height: sectionSpacing / 2),
3908. \_buildFeedbackSection(
3909. context,
3910. isMobile,
3911. cardPadding,
3912. fontSizeSubtitle,
3913. ),
3914. ],
3915. ),
3916. ],
3917. ),
3918. ),
3919. );
3920. },
3921. );
3922. },
3923. );
3924. }
3925. Widget \_buildWelcomeBanner(
3926. BuildContext context,
3927. AuthService authService,
3928. bool isMobile,
3929. double fontSizeTitle,
3930. double fontSizeSubtitle,
3931. ) {
3932. return Card(
3933. elevation: 6,
3934. shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),
3935. child: Container(
3936. decoration: BoxDecoration(
3937. gradient: LinearGradient(
3938. colors: [Colors.indigo[700]!, Colors.indigo[500]!],
3939. begin: Alignment.topLeft,
3940. end: Alignment.bottomRight,
3941. ),
3942. borderRadius: BorderRadius.circular(12),
3943. ),
3944. padding: EdgeInsets.all(isMobile ? 12.0 : 16.0),
3945. child: Column(
3946. mainAxisSize: MainAxisSize.min,
3947. crossAxisAlignment: CrossAxisAlignment.center,
3948. children: [
3949. Row(
3950. mainAxisAlignment: MainAxisAlignment.center,
3951. children: [
3952. Text(
3953. 'Welcome${authService.isLoggedIn ? ', ${authService.username}' : ''}!',
3954. style: GoogleFonts.poppins(
3955. color: Colors.white,
3956. fontSize: fontSizeTitle,
3957. fontWeight: FontWeight.w600,
3958. ),
3959. textAlign: TextAlign.center,
3960. ),
3961. SizedBox(width: isMobile ? 8.0 : 12.0),
3962. Icon(
3963. Icons.favorite,
3964. color: Colors.white,
3965. size: isMobile ? 28.0 : 32.0,
3966. ),
3967. ],
3968. ),
3969. SizedBox(height: 8),
3970. Text(
3971. 'Your health, our priority. Manage your care with ease.',
3972. style: GoogleFonts.poppins(
3973. color: Colors.white.withOpacity(0.9),
3974. fontSize: fontSizeSubtitle,
3975. ),
3976. textAlign: TextAlign.center,
3977. ),
3978. ],
3979. ),
3980. ),
3981. );
3982. }
3983. Widget \_buildLoginPrompt(
3984. BuildContext context,
3985. bool isMobile,
3986. double cardPadding,
3987. double fontSizeTitle,
3988. ) {
3989. return Card(
3990. elevation: 4,
3991. shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),
3992. child: Padding(
3993. padding: EdgeInsets.all(cardPadding),
3994. child: Column(
3995. mainAxisAlignment: MainAxisAlignment.center,
3996. children: [
3997. Icon(
3998. Icons.lock,
3999. size: isMobile ? 28.0 : 32.0,
4000. color: Colors.indigo[700],
4001. ),
4002. SizedBox(height: 12),
4003. Text(
4004. 'Please login or sign up to view your dashboard.',
4005. style: GoogleFonts.poppins(
4006. fontSize: fontSizeTitle,
4007. fontWeight: FontWeight.w500,
4008. color: const Color(0xFF1F2937),
4009. ),
4010. textAlign: TextAlign.center,
4011. ),
4012. SizedBox(height: 16),
4013. ElevatedButton(
4014. onPressed: () {
4015. showDialog(
4016. context: context,
4017. builder: (context) => const AuthDialog(),
4018. );
4019. },
4020. child: Text('Login / Sign Up'),
4021. style: ElevatedButton.styleFrom(
4022. padding: EdgeInsets.symmetric(
4023. horizontal: isMobile ? 20.0 : 24.0,
4024. vertical: isMobile ? 10.0 : 12.0,
4025. ),
4026. ),
4027. ),
4028. ],
4029. ),
4030. ),
4031. );
4032. }
4033. Widget \_buildUserProfile(
4034. BuildContext context,
4035. AuthService authService,
4036. bool isMobile,
4037. double cardPadding,
4038. double fontSizeTitle,
4039. double fontSizeSubtitle,
4040. ) {
4041. return Card(
4042. elevation: 4,
4043. shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),
4044. child: Padding(
4045. padding: EdgeInsets.all(cardPadding),
4046. child: Row(
4047. children: [
4048. CircleAvatar(
4049. backgroundColor: Colors.indigo.withOpacity(0.1),
4050. child: Icon(
4051. Icons.person,
4052. color: Colors.indigo[700],
4053. size: isMobile ? 24.0 : 28.0,
4054. ),
4055. radius: isMobile ? 24.0 : 28.0,
4056. ),
4057. SizedBox(width: 12.0),
4058. Expanded(
4059. child: Column(
4060. crossAxisAlignment: CrossAxisAlignment.start,
4061. children: [
4062. Text(
4063. authService.username ?? 'User',
4064. style: GoogleFonts.poppins(
4065. fontSize: fontSizeTitle,
4066. fontWeight: FontWeight.w600,
4067. color: const Color(0xFF1F2937),
4068. ),
4069. ),
4070. Text(
4071. authService.email ?? 'N/A',
4072. style: GoogleFonts.poppins(
4073. fontSize: fontSizeSubtitle,
4074. color: Colors.grey[600],
4075. ),
4076. ),
4077. ],
4078. ),
4079. ),
4080. OutlinedButton.icon(
4081. onPressed: () {
4082. authService.logout();
4083. final tabController = DefaultTabController.of(context);
4084. if (tabController != null) {
4085. tabController.animateTo(0);
4086. (context as Element).markNeedsBuild();
4087. }
4088. },
4089. icon: Icon(Icons.logout, size: isMobile ? 16.0 : 18.0),
4090. label: Text('Logout'),
4091. style: OutlinedButton.styleFrom(
4092. padding: EdgeInsets.symmetric(
4093. horizontal: isMobile ? 12.0 : 16.0,
4094. vertical: isMobile ? 6.0 : 8.0,
4095. ),
4096. ),
4097. ),
4098. ],
4099. ),
4100. ),
4101. );
4102. }
4103. Widget \_buildOverviewCharts(
4104. BuildContext context,
4105. bool isMobile,
4106. bool isTablet,
4107. double cardPadding,
4108. double chartHeight,
4109. double fontSizeSubtitle,
4110. double screenWidth,
4111. ) {
4112. final int crossAxisCount = isMobile
4113. ? 1
4114. : isTablet
4115. ? 2
4116. : 3;
4117. return Card(
4118. elevation: 4,
4119. shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),
4120. child: Padding(
4121. padding: EdgeInsets.all(cardPadding),
4122. child: GridView.count(
4123. shrinkWrap: true,
4124. physics: const NeverScrollableScrollPhysics(),
4125. crossAxisCount: crossAxisCount,
4126. crossAxisSpacing: isMobile ? 12.0 : 16.0,
4127. mainAxisSpacing: isMobile ? 12.0 : 16.0,
4128. childAspectRatio: isMobile
4129. ? 1.2
4130. : isTablet
4131. ? 1.1
4132. : 1.0,
4133. children: \_buildChartWidgets(
4134. context,
4135. isMobile,
4136. chartHeight,
4137. fontSizeSubtitle,
4138. ),
4139. ),
4140. ),
4141. );
4142. }
4143. List<Widget> \_buildChartWidgets(
4144. BuildContext context,
4145. bool isMobile,
4146. double chartHeight,
4147. double fontSizeSubtitle,
4148. ) {
4149. return [
4150. \_buildChartCard(
4151. context: context,
4152. isMobile: isMobile,
4153. title: 'Appointments by Department',
4154. chart: \_buildDepartmentChart(isMobile, chartHeight),
4155. fontSizeSubtitle: fontSizeSubtitle,
4156. ),
4157. \_buildChartCard(
4158. context: context,
4159. isMobile: isMobile,
4160. title: 'Patient Satisfaction',
4161. chart: \_buildSatisfactionChart(isMobile, chartHeight),
4162. fontSizeSubtitle: fontSizeSubtitle,
4163. ),
4164. \_buildChartCard(
4165. context: context,
4166. isMobile: isMobile,
4167. title: 'Appointment Status',
4168. chart: \_buildStatusChart(isMobile, chartHeight),
4169. fontSizeSubtitle: fontSizeSubtitle,
4170. ),
4171. ];
4172. }
4173. Widget \_buildChartCard({
4174. required BuildContext context,
4175. required bool isMobile,
4176. required String title,
4177. required Widget chart,
4178. required double fontSizeSubtitle,
4179. }) {
4180. return Column(
4181. crossAxisAlignment: CrossAxisAlignment.start,
4182. children: [
4183. Text(
4184. title,
4185. style: GoogleFonts.poppins(
4186. fontSize: fontSizeSubtitle,
4187. fontWeight: FontWeight.w600,
4188. color: const Color(0xFF1F2937),
4189. ),
4190. ),
4191. SizedBox(height: 8),
4192. Container(
4193. height: isMobile ? 180.0 : 220.0,
4194. decoration: BoxDecoration(
4195. color: Colors.white,
4196. borderRadius: BorderRadius.circular(12),
4197. boxShadow: [
4198. BoxShadow(
4199. color: Colors.black.withOpacity(0.05),
4200. blurRadius: 8,
4201. offset: const Offset(0, 2),
4202. ),
4203. ],
4204. ),
4205. padding: const EdgeInsets.all(12),
4206. child: chart,
4207. ),
4208. ],
4209. );
4210. }
4211. Widget \_buildDepartmentChart(bool isMobile, double chartHeight) {
4212. return SizedBox(
4213. height: chartHeight,
4214. child: BarChart(
4215. BarChartData(
4216. alignment: BarChartAlignment.spaceAround,
4217. maxY: 200,
4218. barTouchData: BarTouchData(
4219. enabled: true,
4220. touchTooltipData: BarTouchTooltipData(
4221. tooltipMargin: 10,
4222. getTooltipItem: (group, groupIndex, rod, rodIndex) {
4223. const titles = ['Cardio', 'Neuro', 'Ortho', 'Pedia', 'Derma'];
4224. return BarTooltipItem(
4225. '${titles[group.x]}\n${rod.toY.toInt()}',
4226. GoogleFonts.poppins(
4227. color: Colors.white,
4228. fontSize: isMobile ? 10 : 12,
4229. fontWeight: FontWeight.w500,
4230. ),
4231. textAlign: TextAlign.center,
4232. children: [],
4233. );
4234. },
4235. getTooltipColor: (group) => Colors.indigo[700]!.withOpacity(0.8),
4236. ),
4237. ),
4238. titlesData: FlTitlesData(
4239. show: true,
4240. bottomTitles: AxisTitles(
4241. sideTitles: SideTitles(
4242. showTitles: true,
4243. getTitlesWidget: (value, meta) {
4244. const titles = ['Cardio', 'Neuro', 'Ortho', 'Pedia', 'Derma'];
4245. return Text(
4246. titles[value.toInt()],
4247. style: GoogleFonts.poppins(
4248. fontSize: isMobile ? 8 : 10,
4249. color: Colors.grey[600],
4250. ),
4251. );
4252. },
4253. ),
4254. ),
4255. leftTitles: AxisTitles(
4256. sideTitles: SideTitles(
4257. showTitles: true,
4258. interval: 50,
4259. getTitlesWidget: (value, meta) {
4260. return Text(
4261. value.toInt().toString(),
4262. style: GoogleFonts.poppins(
4263. fontSize: isMobile ? 8 : 10,
4264. color: Colors.grey[600],
4265. ),
4266. );
4267. },
4268. ),
4269. ),
4270. topTitles: const AxisTitles(
4271. sideTitles: SideTitles(showTitles: false),
4272. ),
4273. rightTitles: const AxisTitles(
4274. sideTitles: SideTitles(showTitles: false),
4275. ),
4276. ),
4277. borderData: FlBorderData(show: false),
4278. barGroups: [
4279. BarChartGroupData(
4280. x: 0,
4281. barRods: [
4282. BarChartRodData(
4283. toY: 120,
4284. color: Colors.indigo[700],
4285. width: isMobile ? 8 : 12,
4286. borderRadius: BorderRadius.circular(4),
4287. ),
4288. ],
4289. ),
4290. BarChartGroupData(
4291. x: 1,
4292. barRods: [
4293. BarChartRodData(
4294. toY: 80,
4295. color: Colors.indigo[600],
4296. width: isMobile ? 8 : 12,
4297. borderRadius: BorderRadius.circular(4),
4298. ),
4299. ],
4300. ),
4301. BarChartGroupData(
4302. x: 2,
4303. barRods: [
4304. BarChartRodData(
4305. toY: 100,
4306. color: Colors.indigo[500],
4307. width: isMobile ? 8 : 12,
4308. borderRadius: BorderRadius.circular(4),
4309. ),
4310. ],
4311. ),
4312. BarChartGroupData(
4313. x: 3,
4314. barRods: [
4315. BarChartRodData(
4316. toY: 150,
4317. color: Colors.indigo[400],
4318. width: isMobile ? 8 : 12,
4319. borderRadius: BorderRadius.circular(4),
4320. ),
4321. ],
4322. ),
4323. BarChartGroupData(
4324. x: 4,
4325. barRods: [
4326. BarChartRodData(
4327. toY: 60,
4328. color: Colors.indigo[300],
4329. width: isMobile ? 8 : 12,
4330. borderRadius: BorderRadius.circular(4),
4331. ),
4332. ],
4333. ),
4334. ],
4335. ),
4336. ),
4337. );
4338. }
4339. Widget \_buildSatisfactionChart(bool isMobile, double chartHeight) {
4340. return SizedBox(
4341. height: chartHeight,
4342. child: BarChart(
4343. BarChartData(
4344. alignment: BarChartAlignment.spaceAround,
4345. maxY: 5,
4346. barTouchData: BarTouchData(
4347. enabled: true,
4348. touchTooltipData: BarTouchTooltipData(
4349. tooltipMargin: 10,
4350. getTooltipItem: (group, groupIndex, rod, rodIndex) {
4351. const titles = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'];
4352. return BarTooltipItem(
4353. '${titles[group.x]}\n${rod.toY.toStringAsFixed(1)}',
4354. GoogleFonts.poppins(
4355. color: Colors.white,
4356. fontSize: isMobile ? 10 : 12,
4357. fontWeight: FontWeight.w500,
4358. ),
4359. textAlign: TextAlign.center,
4360. children: [],
4361. );
4362. },
4363. getTooltipColor: (group) => Colors.indigo[700]!.withOpacity(0.8),
4364. ),
4365. ),
4366. titlesData: FlTitlesData(
4367. show: true,
4368. bottomTitles: AxisTitles(
4369. sideTitles: SideTitles(
4370. showTitles: true,
4371. getTitlesWidget: (value, meta) {
4372. const titles = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'];
4373. return Text(
4374. titles[value.toInt()],
4375. style: GoogleFonts.poppins(
4376. fontSize: isMobile ? 8 : 10,
4377. color: Colors.grey[600],
4378. ),
4379. );
4380. },
4381. ),
4382. ),
4383. leftTitles: AxisTitles(
4384. sideTitles: SideTitles(
4385. showTitles: true,
4386. interval: 1,
4387. getTitlesWidget: (value, meta) {
4388. return Text(
4389. value.toInt().toString(),
4390. style: GoogleFonts.poppins(
4391. fontSize: isMobile ? 8 : 10,
4392. color: Colors.grey[600],
4393. ),
4394. );
4395. },
4396. ),
4397. ),
4398. topTitles: const AxisTitles(
4399. sideTitles: SideTitles(showTitles: false),
4400. ),
4401. rightTitles: const AxisTitles(
4402. sideTitles: SideTitles(showTitles: false),
4403. ),
4404. ),
4405. borderData: FlBorderData(show: false),
4406. barGroups: [
4407. BarChartGroupData(
4408. x: 0,
4409. barRods: [
4410. BarChartRodData(
4411. toY: 4.2,
4412. color: Colors.indigo[700],
4413. width: isMobile ? 8 : 12,
4414. borderRadius: BorderRadius.circular(4),
4415. ),
4416. ],
4417. ),
4418. BarChartGroupData(
4419. x: 1,
4420. barRods: [
4421. BarChartRodData(
4422. toY: 4.3,
4423. color: Colors.indigo[600],
4424. width: isMobile ? 8 : 12,
4425. borderRadius: BorderRadius.circular(4),
4426. ),
4427. ],
4428. ),
4429. BarChartGroupData(
4430. x: 2,
4431. barRods: [
4432. BarChartRodData(
4433. toY: 4.5,
4434. color: Colors.indigo[500],
4435. width: isMobile ? 8 : 12,
4436. borderRadius: BorderRadius.circular(4),
4437. ),
4438. ],
4439. ),
4440. BarChartGroupData(
4441. x: 3,
4442. barRods: [
4443. BarChartRodData(
4444. toY: 4.1,
4445. color: Colors.indigo[400],
4446. width: isMobile ? 8 : 12,
4447. borderRadius: BorderRadius.circular(4),
4448. ),
4449. ],
4450. ),
4451. BarChartGroupData(
4452. x: 4,
4453. barRods: [
4454. BarChartRodData(
4455. toY: 4.4,
4456. color: Colors.indigo[300],
4457. width: isMobile ? 8 : 12,
4458. borderRadius: BorderRadius.circular(4),
4459. ),
4460. ],
4461. ),
4462. BarChartGroupData(
4463. x: 5,
4464. barRods: [
4465. BarChartRodData(
4466. toY: 4.6,
4467. color: Colors.indigo[200],
4468. width: isMobile ? 8 : 12,
4469. borderRadius: BorderRadius.circular(4),
4470. ),
4471. ],
4472. ),
4473. ],
4474. ),
4475. ),
4476. );
4477. }
4478. Widget \_buildStatusChart(bool isMobile, double chartHeight) {
4479. return SizedBox(
4480. height: chartHeight,
4481. child: BarChart(
4482. BarChartData(
4483. alignment: BarChartAlignment.spaceAround,
4484. maxY: 250,
4485. barTouchData: BarTouchData(
4486. enabled: true,
4487. touchTooltipData: BarTouchTooltipData(
4488. tooltipMargin: 10,
4489. getTooltipItem: (group, groupIndex, rod, rodIndex) {
4490. const titles = ['Upcoming', 'Completed', 'Cancelled'];
4491. return BarTooltipItem(
4492. '${titles[group.x]}\n${rod.toY.toInt()}',
4493. GoogleFonts.poppins(
4494. color: Colors.white,
4495. fontSize: isMobile ? 10 : 12,
4496. fontWeight: FontWeight.w500,
4497. ),
4498. textAlign: TextAlign.center,
4499. children: [],
4500. );
4501. },
4502. getTooltipColor: (group) => Colors.indigo[700]!.withOpacity(0.8),
4503. ),
4504. ),
4505. titlesData: FlTitlesData(
4506. show: true,
4507. bottomTitles: AxisTitles(
4508. sideTitles: SideTitles(
4509. showTitles: true,
4510. getTitlesWidget: (value, meta) {
4511. const titles = ['Upcoming', 'Completed', 'Cancelled'];
4512. return Text(
4513. titles[value.toInt()],
4514. style: GoogleFonts.poppins(
4515. fontSize: isMobile ? 8 : 10,
4516. color: Colors.grey[600],
4517. ),
4518. );
4519. },
4520. ),
4521. ),
4522. leftTitles: AxisTitles(
4523. sideTitles: SideTitles(
4524. showTitles: true,
4525. interval: 50,
4526. getTitlesWidget: (value, meta) {
4527. return Text(
4528. value.toInt().toString(),
4529. style: GoogleFonts.poppins(
4530. fontSize: isMobile ? 8 : 10,
4531. color: Colors.grey[600],
4532. ),
4533. );
4534. },
4535. ),
4536. ),
4537. topTitles: const AxisTitles(
4538. sideTitles: SideTitles(showTitles: false),
4539. ),
4540. rightTitles: const AxisTitles(
4541. sideTitles: SideTitles(showTitles: false),
4542. ),
4543. ),
4544. borderData: FlBorderData(show: false),
4545. barGroups: [
4546. BarChartGroupData(
4547. x: 0,
4548. barRods: [
4549. BarChartRodData(
4550. toY: 50,
4551. color: Colors.indigo[700],
4552. width: isMobile ? 8 : 12,
4553. borderRadius: BorderRadius.circular(4),
4554. ),
4555. ],
4556. ),
4557. BarChartGroupData(
4558. x: 1,
4559. barRods: [
4560. BarChartRodData(
4561. toY: 200,
4562. color: Colors.indigo[600],
4563. width: isMobile ? 8 : 12,
4564. borderRadius: BorderRadius.circular(4),
4565. ),
4566. ],
4567. ),
4568. BarChartGroupData(
4569. x: 2,
4570. barRods: [
4571. BarChartRodData(
4572. toY: 30,
4573. color: Colors.indigo[500],
4574. width: isMobile ? 8 : 12,
4575. borderRadius: BorderRadius.circular(4),
4576. ),
4577. ],
4578. ),
4579. ],
4580. ),
4581. ),
4582. );
4583. }
4584. Widget \_buildFeedbackSection(
4585. BuildContext context,
4586. bool isMobile,
4587. double cardPadding,
4588. double fontSizeSubtitle,
4589. ) {
4590. return FutureBuilder<List<Map<String, dynamic>>>(
4591. future: Provider.of<FeedbackService>(context).loadFeedback(),
4592. builder: (context, snapshot) {
4593. if (snapshot.connectionState == ConnectionState.waiting) {
4594. return const Center(child: CircularProgressIndicator());
4595. }
4596. if (snapshot.hasError) {
4597. return Card(
4598. elevation: 4,
4599. shape: RoundedRectangleBorder(
4600. borderRadius: BorderRadius.circular(12),
4601. ),
4602. child: Padding(
4603. padding: EdgeInsets.all(cardPadding),
4604. child: Center(
4605. child: Text(
4606. 'Error loading feedback',
4607. style: GoogleFonts.poppins(
4608. fontSize: fontSizeSubtitle,
4609. color: const Color(0xFF1F2937),
4610. ),
4611. ),
4612. ),
4613. ),
4614. );
4615. }
4616. final feedbackList = snapshot.data ?? [];
4617. if (feedbackList.isEmpty) {
4618. return Card(
4619. elevation: 4,
4620. shape: RoundedRectangleBorder(
4621. borderRadius: BorderRadius.circular(12),
4622. ),
4623. child: Padding(
4624. padding: EdgeInsets.all(cardPadding),
4625. child: Center(
4626. child: Text(
4627. 'No feedback submitted yet.',
4628. style: GoogleFonts.poppins(
4629. fontSize: fontSizeSubtitle,
4630. color: const Color(0xFF1F2937),
4631. ),
4632. ),
4633. ),
4634. ),
4635. );
4636. }
4637. return ListView.separated(
4638. shrinkWrap: true,
4639. physics: const NeverScrollableScrollPhysics(),
4640. itemCount: feedbackList.length,
4641. separatorBuilder: (context, index) => const SizedBox(height: 12),
4642. itemBuilder: (context, index) {
4643. final feedback = feedbackList[index];
4644. return Card(
4645. elevation: 4,
4646. shape: RoundedRectangleBorder(
4647. borderRadius: BorderRadius.circular(12),
4648. ),
4649. child: Padding(
4650. padding: EdgeInsets.all(isMobile ? 10.0 : 12.0),
4651. child: Column(
4652. crossAxisAlignment: CrossAxisAlignment.start,
4653. children: [
4654. Row(
4655. children: [
4656. Icon(
4657. Icons.star,
4658. color: Colors.amber,
4659. size: isMobile ? 18.0 : 20.0,
4660. ),
4661. const SizedBox(width: 8),
4662. Text(
4663. 'Rating: ${feedback['rating'].toStringAsFixed(1)}',
4664. style: GoogleFonts.poppins(
4665. fontSize: fontSizeSubtitle,
4666. fontWeight: FontWeight.w600,
4667. color: const Color(0xFF1F2937),
4668. ),
4669. ),
4670. ],
4671. ),
4672. const SizedBox(height: 8),
4673. Text(
4674. feedback['comment'] ?? 'N/A',
4675. style: GoogleFonts.poppins(
4676. fontSize: fontSizeSubtitle,
4677. color: Colors.grey[600],
4678. ),
4679. ),
4680. const SizedBox(height: 8),
4681. Text(
4682. 'Submitted on: ${feedback['date'] ?? 'N/A'}',
4683. style: GoogleFonts.poppins(
4684. fontSize: isMobile ? 10.0 : 12.0,
4685. color: Colors.grey[600],
4686. ),
4687. ),
4688. ],
4689. ),
4690. ),
4691. );
4692. },
4693. );
4694. },
4695. );
4696. }
4697. }